

Vehicle Routing Problems in Signalized Traffic Networks

A DISSERTATION
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Jie Sun

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
Doctor of Philosophy

Advisor: Henry X. Liu

August, 2014

© Jie Sun 2014

ALL RIGHTS RESERVED

Acknowledgements

I would like to first thank my advisor Henry X. Liu, who has guided me through all the difficulties during my Ph.D study. His excellence in work has set a great model for me to follow. This dissertation is not even possible without his generous support throughout my Ph.D study.

And I am also grateful to the professors here in Minnesota, who taught me a lot during my Ph.D study. I would like to thank Professor Nikolas Geroliminis, who has provided me the first opportunity to study in the U.S. His willingness to teach and help made my first year study in a new country easy and enjoyable. His guidance greatly increased my altitude not only from academical aspects but also from many other aspects. Professor Gary Davis taught me a lot of advanced mathematics that is crucial in my research. Professor David Levinson extended my knowledge to broad areas of transportation and helped to learn various ways of looking at problems. I would also like to give special thanks to Professor Shashi Shekhar and Dr. John Hourdos for their service on my committee and great suggestions on my dissertation.

My study here also allowed me to have the opportunity to work with many amazing fellow students. I would like thank Dr. Saif Eddin Jabari, Dr. Heng Hu, Dr. Xiaozheng He, Dr. Xinkai Wu, Dr. Xuan Di, Indrajit Chatterjee and Jianfeng Zheng for their useful suggestions to my dissertation and help on my

research. And my gratitude should also be extended to other students including Shengyin Shen, Shihong Huang, Hui Xiong, Feili Hong, and Carlos J Carrión, who helped me a lot during the study.

I am also grateful to all my friends who have made my life in the U.S. enjoyable. Finally, but most importantly, my gratitude goes to my family. I want to especially thank my wife, Yuanyuan Luo, for her love and patience throughout the years. I want to thank my cat, Xiao Mao, for the comfort and happiness she brought to me. I am also grateful to my parents, Xian Sun and Guoping Xu, who have made all of these possible and always encourage me to pursue my goals.

Dedication

To my parents and my wife.

Abstract

This dissertation studied various path search problems when traffic signal information and traffic state is explicitly considered. The research is motivated by the increasing availability of high-resolution traffic data including signal information, which is seldom available in the past. In order to properly account for the randomness resulting from vehicle-actuated traffic signals and the correlation from signal coordination, the theory of Markov decision process (MDP) is used. By taking advantage of the cyclic property of traffic signals, the problem is formulated as an infinite horizon and finite state space MDP with absorbing state set. The objective is to find the optimal policy that gives the minimum expected total cost to the destination.

The state space of the problem is generated based on underlying traffic network geometry and signal control information. Delay distributions at intersections together with signal control parameters, such as cycle length and offset, are used to construct the transition probabilities between states. It will be shown that the required delay distributions can be estimated from readily available field traffic data. The problem where the cost is travel time is first studied. When the cost of concern is the travel time, it includes intersection delays and link travel times. Value iteration method is used to solve the MDP problem when there is only one cost of concern.

In addition, the problem whose cost of concern is environmentally related is also studied. Vehicle trajectories are estimated based on traffic signal information and queuing dynamics at intersections, and put into microscopic vehicle emission models, the results from which are used to calculate the environmental costs for

the path search problem. When multiple costs of concern present, the problem is formulated as a constrained MDP problem. Linear programming formulation of MDP is introduced to solve constrained MDP problem. The proposed methods are tested in a hypothetical traffic network, as well as a real world traffic network in the City of Pasadena, CA.

Contents

Acknowledgements	i
Dedication	iii
Abstract	iv
List of Tables	ix
List of Figures	x
1 Introduction	1
1.1 Problem statement	1
1.2 Research contributions	5
1.2.1 Development of a vehicle routing model considering traffic signal and queuing delays	5
1.2.2 Development of a path search method with environmental objectives	6
1.2.3 Formulation of a path search problem considering multiple objectives	7
1.3 Dissertation organization	8

2	Background	10
2.1	Shortest path problem (SPP)	10
2.2	Markov decision process	11
2.3	Eco-routing problem	13
2.4	Traffic signal control and high-resolution traffic data	16
3	Stochastic Vehicle Routing Problems Considering Traffic Signals	19
3.1	Problem specifications	20
3.2	Markov decision process (MDP)	21
3.3	Model formulation	23
3.3.1	State space and action	23
3.3.2	Time discretization	26
3.3.3	Cost function and transition probability	31
3.4	Incorporation of real-time traffic signals	35
3.5	Estimation of intersection delay distributions	36
3.6	Value iteration method	42
4	Stochastic Eco-routing Problems Considering Traffic Signals	46
4.1	Problem specifications	46
4.2	Model formulation	47
4.3	Estimation of eco-costs	49
4.4	The constrained eco-routing problem	58
5	Numerical Examples	64
5.1	A hypothetical signalized traffic network	64
5.1.1	Network layout	64
5.1.2	Expected total travel time as objective	65

5.1.3	Expected fuel consumption as objective	69
5.2	A Real world signalized traffic network	75
5.2.1	Test site settings	75
5.2.2	Expected total travel time as objective	79
5.2.3	CO emission as objective	84
5.2.4	The constrained problem	86
6	Summary and Future Research	89
6.1	Summary	89
6.2	Future research	92
	References	95
	Appendix A. Notations	102

List of Tables

2.1	Summary of major vehicle emission models	14
3.1	Second-by-second state transition probability and cost	30
3.2	Aggregated state transition probability and cost	31
3.3	Sample cycle information	38
5.1	Link travel time distribution	66
5.2	Intersection delay distributions	66
5.3	Transition probability at coordinated intersections	72
5.4	Transition probability at non-coordinated intersections	72
5.5	Expected intersection delays (seconds)	73
5.6	Stop probability at intersections	73
5.7	Vehicle characteristics and link segment division	74
5.8	Fuel consumption parameters	74
5.9	Sample signal data	78
5.10	Signal events	79
5.11	Average travel time reduction by using proposed method	83
5.12	CO emission parameters	85
5.13	Optimal values and constraints constant of different costs	87
A.1	Notation list	102

List of Figures

3.1	State transition at intersections with coordination	25
3.2	Two time discretization schemes	27
3.3	Transition probability at intersections with coordination	34
3.4	Sample data collection site	37
3.5	Intersection delays with different queue profiles	38
3.6	Examples of delay distributions	41
4.1	Vehicle status transitions at intersections and on links	51
4.2	Vehicle trajectory types on links	52
4.3	Vehicle trajectories on links with different lengths	53
4.4	A vehicle trajectory on a link	54
5.1	The layout of the hypothetical signalized traffic network	65
5.2	Test results in the hypothetical network	67
5.3	Intersection phase settings	70
5.4	Intersection signal phases and time periods	71
5.5	Optimal fuel consumption policy	74
5.6	The location of test site in the City of Pasadena, CA (Google Maps)	76
5.7	Network configurations	77
5.8	An example of red light delay distribution	80
5.9	Optimal policy to minimize expected total travel time	82

5.10	Time dependent actions at intersection 6	83
5.11	Optimal policy to minimize expected total CO emission	86
5.12	Time dependent actions at intersection 2	87
5.13	Randomized action at intersection 2 at the first cycle time	88

Chapter 1

Introduction

1.1 Problem statement

Path search is a fundamental problem in transportation science. It has usually been formulated as a shortest path problem (SPP) and is the key sub-problem to many other problems such as traffic assignment problem. Traditionally, the link travel time is the major cost under consideration and modeled as static and deterministic. Although methods accounting for the randomness of link travel time have been developed, it appears that the efforts to find the shortest path considering traffic signals are limited. When considered in the literature, they were usually regarded as fixed timing signals and the corresponding shortest path problems were usually solved by variations of label correcting algorithm (Chen and Yang, 2000; Ahuja et al., 2002; Yang and Miller-Hooks, 2004).

In a traffic network, however, not all the intersections are controlled by fixed timing traffic signals, and many major signalized arterials are controlled by vehicle-actuated traffic signals with coordination. Durations of signal phases are no longer

fixed when traffic signals are vehicle actuated, which causes randomness in intersection delays. This is a consequence of random vehicle arrivals at intersections. In addition, signal coordination results in extra correlation between delays at adjacent intersections. The correlation of intersection delays comes from two sources. First, the traffic propagation along road segments can cause delays at intersections correlated. Furthermore, these delays may also be correlated because of the signal coordination used in some urban arterials. These factors make a path search problem considering traffic signals much more complicated.

First of all, it is easy to see that the path search problem is no longer deterministic. The delays at intersection and corresponding costs become random. This requires the use of a stochastic model. Furthermore, the correlation between these delays requires a stochastic model that can handle random variables with correlation. These requirements make it difficult to solve the problem by using traditional shortest path algorithms, such as label correcting algorithm.

Another big challenge to accommodate different types of traffic signals in a path search problem is information collection. Even if we have a stochastic model at hand, we need to provide necessary information to the model as inputs. In the past, such information is limited. Few people collected detailed traffic signal and related information on arterial road networks. Recent development of data collection technology has started to change the situation. For example, the SMART-Signal system developed by the University of Minnesota, Twin Cities, is able to collect and archive high-resolution traffic data (Liu and Ma, 2009; Liu et al., 2009). These data include detailed signal status information as well as vehicle actuation information from inductive loop detectors. This opens a new opportunity to incorporate vehicle-actuated traffic signals in a path search problem.

The key to incorporate vehicle-actuated traffic signals in a path search problem is to develop a model that can deal with randomness and correlation of intersection delays. The formulation of Markov decision process (MDP) naturally incorporates randomness and correlation at the same time. It is known that the general theory of MDP can be used to analyze the stochastic shortest path problem (see Bertsekas (1995) and the references therein). It is unknown, however, how this theory can be used to model the delays caused by different types of signal control strategies.

Recently, environmental impacts of transportation activities have received increasing attention. For most vehicles, each trip will consume certain amount of fuel and emit various types of pollutants, such as CO_2 and NO_x . It will be valuable to consider these costs in a path problem. This is especially true in signalized road networks, because there are much more vehicle speed fluctuations due to signal controls, compared to the trips on freeways. And research has shown that short term vehicle activities such as acceleration and deceleration have significant impacts on vehicle fuel consumption and emission rates (Rakha et al., 2000; Ahn et al., 2002; El-Shawarby et al., 2005).

The major difficulty of considering environmental costs in a path search problem lies in the calculation of related costs, which is more complicated than the calculating time or distance cost. It turns out the use of microscopic vehicle emission models is necessary to calculate environmental costs of vehicle trips (Barth et al., 2000; Frey et al., 2010; Oneyama et al., 2001; Rakha et al., 2004; EPA, 2012). One of the major inputs to these models is second by second vehicle speed profiles. How to obtain such information before the actual happening of the trips is the question that needs to be addressed for environmental related path search problems. This dissertation will develop a method to estimate vehicle trajectories

in signalized road networks based on signal status and vehicle actuation information.

As there are usually more than one environmental costs resulting from vehicle activities, an obvious follow-up question is how to solve a path search problem when there are multiple costs of interest. It would be ideal if one can find a solution that minimizes all the costs at the same time, but this may not always be feasible. One can formulate an multi-objective problem that uses weighted average of different costs as its objective. While this is a viable solution, it may not be the most appropriate one in the context of our problems. On one hand, some of the costs may not be comparable. For example, how one can compare one minute of travel time to one gram of fuel consumption can be a difficult question to answer in itself. This makes it difficult to decide the weights in the objective function. On the other hand, it is not necessary to minimize some of the costs. For example, it may be sufficient to emission of CO_2 below certain targets according to international agreement such as “Kyoto Protocol”.

So in this dissertation, we adopt a constraint approach by adding some of the costs as constraints. Although minimizing the emission of some pollutants, e.g. CO_2 , may not be achieved, the goal is to maintain pollution emission below certain targets while minimizing some primary cost, such as travel time or fuel consumption. We will use a method where we have a primary objective that needs to be minimized, e.g. travel time, and a set of constraints that control other costs of interest which relates to emission targets.

1.2 Research contributions

1.2.1 Development of a vehicle routing model considering traffic signal and queuing delays

In this research, a method to find optimal paths in signalized traffic networks will be developed based on MDP with finite state space, utilizing cyclic property of traffic signals. It explicitly accounts for red light delays caused by different types of traffic signals. It is especially suitable for the situations where vehicle-actuated traffic signals with coordination are used. In addition, delays caused by queuing vehicles are also incorporated. Both delays are allowed to be random and will be dependent on the vehicle arrival times at intersections. The transition probabilities between arrival times at adjacent intersections are estimated using traffic data, signal actuation data, and signal control parameters such as cycle length and offset. The final output of the model is a policy which provides en-route optimal route choices depending the realization of vehicle's locations, arrival times at intersections, and signal status at arrival times.

The key to finding optimal paths in a signalized traffic network is to collect the required information. One of the most important part is to calculate the intersection delays when actuated traffic signal is used. This requires the detailed information about signal status and vehicle arrival information. Since these information was not available in the past, most work on path search problems considering traffic signals assumed fixed timing traffic signal control was used.

In this research, high-resolution traffic data, including detailed signal status and individual vehicle arrival information, are used to estimate intersection delays. The calculation decomposes intersection delay into two parts: red light delay and queuing delay. Red light delay is the delay a vehicle will experience at an

intersection if red light presents when it arrives at that intersection. Queuing delay is the extra delay caused by the queued vehicles in front of current vehicle when it arrives at the intersection. Since these delays are random, they need to be modeled as random variables. Their distributions are estimated using archived data in the past. This is the building block of solving path finding problems in a signalized traffic network. And it will be shown in this dissertation that all the required input information are readily available from existing data collection system.

1.2.2 Development of a path search method with environmental objectives

Instead of travel time or distance, a path search problem can also use other costs in its objective function. This dissertation will solve the path search problems with environmental costs as objectives. For problems where environmental costs are the concerns, the vital part is the calculation of environmental costs, such as fuel consumption, CO₂ emission. Since the actual environmental impacts of vehicle activities depend on many factors, people have developed microscopic vehicle emission models to estimate these costs. The challenge is how to incorporate a microscopic vehicle emission model into a path search problem.

A microscopic vehicle emission model usually uses second by second vehicle speed as one of its inputs. This can be used as the bridge between a microscopic vehicle emission model and a path search algorithm. To obtain second by second vehicle speed information, it has been proposed to use historical GPS-based vehicle trajectory (Ericsson et al., 2006; Hammarström, 1999; Pelkmans et al., 2004) or

microscopic vehicle simulation model (Rakha et al., 2012). The problem of GPS-based approach is its data availability. It requires network wide coverage of GPS data, which may not be available for many places. In addition, most approaches using GPS-based data are based on traditional shortest path algorithm, which is difficult to incorporate valuable information from signal controllers. As for simulation based approach, it would require significant calibration efforts and large computation power at run time. Different from these approaches, we will use signal status and queuing information to estimate vehicle trajectories in a signalized traffic network. As this information can be continuously collected from widespread infrastructures, including signal controllers and inductive loop detectors, this approach seems to be more promising.

1.2.3 Formulation of a path search problem considering multiple objectives

By adding environmental costs into a path search problem, we face a situation where there are more than just one cost of concern. Of course, it would be ideal if there is a solution that minimizes all the costs. But it may not be always possible to achieve this goal. Instead, for example, one may want to minimize the fuel consumption while keeping the travel time within certain amount of time. This motivates us to formulate a path search problem with constraints. Based on the MDP formulation developed in earlier chapters, the last part of the dissertation will utilize the linear programming formulation of MDP. This allows constraints to be added to the problems so that various costs of concerns can be accounted for at the same time.

1.3 Dissertation organization

This dissertation is organized as follows:

- Chapter 2 provides the background of this dissertation. It reviews the basics of shortest path problems and those efforts that considered traffic signals in the literature. It also briefly introduces the Markov decision process that can be used to solve stochastic shortest path problems. Following this, the recent development on path finding problems with environmental objectives is reviewed. Finally, this chapter also gives a overview of traffic signal control operations that are relevant to this dissertation.
- Chapter 3 formulates a vehicle routing problem considering traffic signals based on Markov decision process. It first specifies the problems. Then Markov decision process is introduced. Based on this formulation, a path finding problem is formulated as an infinite-horizon, finite state space MDP with expected total costs as its optimality criteria. Next, the chapter discusses how to incorporate real-time traffic signal status into the formulation when this information is available. Different time schemes and the conversion between them are also discussed this chapter. Following this, the way to get intersection delay distributions from high-resolution traffic data is demonstrated. These distributions are needed to construct transition probabilities and calculate expected travel time costs. At the end of this chapter, value iteration method is introduced to solve the problem. Value iteration method is widely used for solving MDP. Analysis of the algorithm is also presented.
- Chapter 4 studies the eco-routing problem considering traffic signals. An eco-routing problem is a vehicle routing problem with vehicle emissions or

fuel consumption as objectives. The problem is still formulated based on MDP but requires the calculation of environmental related costs in a path search problem. Microscopic vehicle emission models are used to calculate environmental related costs. And a method is developed to estimate vehicle trajectories in a signalized traffic network based on information at intersections. The estimated trajectories will be used as the inputs to microscopic vehicle emission models for the calculation of environmental costs. This chapter then studies the constrained eco-routing problem by introducing a linear programming formulation of MDP. This formulation allows constraints to be added to the problem so that multiple costs can be simultaneously considered.

- Chapter 5 gives numerical examples of the proposed models. First, a hypothetical signalized network with 12 intersections will be used as a demonstration. It shows how the required information looks like and what's the final outputs of the algorithm. Then, the proposed methods will be tested on a real network from City of Pasadena, California, USA. This example involves a larger network and much higher data requirements. It shows the proposed methods are readily applicable to real world problems.
- Chapter 6 summaries this dissertation and discusses about the future directions of this research.

Chapter 2

Background

2.1 Shortest path problem (SPP)

In this research, the problems we are dealing with have their roots in shortest path problems, which have been studied for decades. In a shortest path problem, distance or travel time was usually the cost of interest. Furthermore, people are more concerned about the costs on links, and these costs were usually modeled as static and deterministic. Algorithms like label-setting algorithm (Dijkstra, 1959) and the label-correcting algorithm (Ford, 1956) were developed to efficiently find the shortest path in this setting.

Besides costs incurred along links, the costs to pass nodes can be also important in a shortest path problem. Inspired by the idea from Caldwell (1961), Kirby and Potts (1969) provided a mathematical formulation, in which turning penalties together with link costs were considered, but no algorithm or implementation was provided. Easa (1985) proposed a method that accounted for prohibited movements but his method could not handle intersection movements with penalties. Based on the research by Kirby and Potts (1969), Ziliaskopoulos and Mahmassani

(1996) proposed an improved label correcting algorithm, where static deterministic turning penalties were included.

Chen and Yang (2000) studied the shortest path problem in the presence of fixed timing traffic signals, which were modeled as multiple time windows. Ahuja et al. (2002) extended the shortest path problem considering fixed timing traffic signals to allow the costs to be time-dependent. They have showed that the minimum time path problem could be solved in polynomial time, but the minimum cost path problems were generally NP-hard. Yang and Miller-Hooks (2004) studied the shortest path problem with adaptive traffic signals. The available and unavailable times for a movement at an intersection were assumed to be exponentially distributed and modeled as a two-state continuous time Markov chain (CTMC). Their paper provided some explanations for the choice of the CTMC modeling approach but no empirical justifications were provided and it is difficult to connect their model to real world traffic signal control parameters.

2.2 Markov decision process

A generalization of the deterministic shortest path problem is that, out of a set of possible distributions, one selects a possible distribution over all possible successive nodes at each node. For a given origin node, the objective is to reach the destination node with probability 1 and minimum expected length. For a given selection of distributions and a given origin node, the path traversed and its length becomes random (Bertsekas and Tsitsiklis, 1991). It is usually assumed that target states or destinations are absorbing and cost-free.

This problem was first formulated by Eaton and Zadeh (1962) as a *problem of pursuit*, whose objective was to intercept a target that moves in finite number of

states with minimum expected time. It has been shown that the problem can be formulated as the one with a stationary target. The notion of proper policy was introduced by Eaton and Zadeh (1962). A policy is called proper if the associated expected cost is finite.

In the literature, this class of problems were also referred as *first passage problem* Derman (1970), *free-time problem* Kushner (1971), *transient programming* Whittle (1983), as well as *stochastic shortest path problem* Bertsekas and Tsitsiklis (1991). Eaton and Zadeh (1962) assumed finite control sets and positive one-stage expected cost except for the destination. Kushner (1971) extended the problem to allow the set of controls at each state to be infinite but required compactness. Bertsekas (1987) relaxed the positive cost assumption of the finite-state and finite-control problem to nonnegative costs. Bertsekas and Tsitsiklis (1991) improved the model so that negative stage costs were allowed in addition to positive stage costs.

In the 1960s, people started to explore the possibility of use linear programming techniques for solving the problems of Markov decision process with different optimality criteria. Linear programming approach has been applied to the problem of Markov decision process with discounted total cost criterion (d'Epenoux, 1960, 1963) and with average cost criterion (Manne, 1960). For the problems studied in this dissertation, we are more interested in the case where expected total cost is used as optimality criteria but the discount factor equals to one. In this case, there is usually leakage out of the system or some sort of absorbing states. The linear programming treatment of the MDP problem has been extensively studied by Kallenberg (1983).

The linear programming formulation of MDP is particularly suitable for the problems with constraints. This provides us an useful approach when we have

more than one cost of interest. In this case, a primary cost can be minimized while other costs are bounded by some given values.

2.3 Eco-routing problem

There can be other costs that are related to vehicle activities in addition to distance and time. Transportation system is a major contributor of air pollution in urban areas. In the U.S., 30% of the nation's total petroleum consumption is made by vehicles (EPA, 2008a) and vehicles produced about one third of carbon dioxide (CO_2) (EPA, 2008b).

Recently, finding the route that is most environmentally friendly has been formulated as “eco-routing” problems and solution methods to these problems have been proposed (Barth et al., 2007; Boriboonsomsin et al., 2012; Ericsson et al., 2006). It has been shown that a time or distance minimizing route does not always minimize fuel consumption or emissions (Ahn and Rakha, 2008; Boriboonsomsin et al., 2012). The calculation of environmental related costs is much more complicated compared to that using time or distance as costs.

Vehicle fuel consumption and emissions depend on many factors. For individual vehicles, microscopic vehicle emission models have been developed to estimate vehicle fuel consumption and emissions (Barth et al., 2000; Frey et al., 2010; Oneyama et al., 2001; Rakha et al., 2004; EPA, 2012) (see Table 2.1). These models usually use second-by-second vehicle speed profiles as one of the most important inputs and calculate vehicle fuel consumption and emission rates as outputs. Second-by-second vehicle speed profiles are equivalent to vehicle trajectories, which are highly dependent on traffic states. The stochastic nature of traffic states makes it difficult to obtain vehicle trajectories for vehicle guidance problems,

which require the trajectories before actual trips. Of course, vehicle emission models also require other inputs such as road grades and vehicle characteristics. But such information is usually static and relatively easy to obtain.

Table 2.1: Summary of major vehicle emission models

Model	Developed by	Outputs
CMEM	University of California, Riverside	HC, CO, NO _x , CO ₂ and fuel
VSP	North Carolina State University	HC, CO, NO, and fuel
VT-Micro	Virginia Tech	HC, CO, NO _x , CO ₂ and fuel
MOVES 2010b	US EPA	CO, NO, NO _x , PM ₁₀ , PM _{2.5} , fuel, etc

So the key to an eco-routing problem is to estimate the vehicle trajectories. When dealing with eco-routing problem, people usually estimate the trajectory on a link based on collected GPS-based vehicle trajectories or a set of explanatory variables for a link. Then, this information is used as the input to vehicle emission models to calculate the vehicles fuel consumption and emissions for that link. After calculating the environmental cost for each link, a standard shortest path algorithm is used to calculate the optimal path that minimizes the environmental impacts.

Ericsson et al. (2006) developed a method for optimizing route choice for lowest fuel consumption. Although the method was based on vehicle simulation models that were capable of calculating fuel consumption and emissions for individual vehicles given detailed speed profiles (Hammarström, 1999; Pelkmans et al., 2004), only fuel consumption was considered by calculating fuel consumption factors on street links using GPS data in that research.

In addition to GPS data, other sources of traffic data, such as loop detector data and probe vehicle data, can also be utilized to calculate fuel consumption and emissions based on data fusion techniques. Researchers from University of California, Riverside, have developed a vehicle navigation system based on CMEM (Barth et al., 2000). Their system provided an optimal route with regards to fuel consumption or various emissions (Barth et al., 2007; Boriboonsomsin et al., 2012).

Different from the GPS data based approach, the eco-routing problem can also be solved using a simulation based approach. Rakha et al. (2012) used a microscopic traffic assignment and simulation software to generate speed and acceleration data. Then, these information was put into a microscopic vehicle emission model called VT-Micro (Rakha et al., 2004) to calculate related costs. Finally, the optimal route was found by solving a traffic assignment problem.

The aforementioned work formulates the eco-routing problem as a single objective problem, although there are various pollutants related vehicle activities. A single objective problem means users can ask for an optimal path with a given objective, e.g. highest fuel efficiency or lowest emission for a pollutant such as CO₂, without considering the costs other than the one used in the objective function. To address this issue, one can formulate a multi-objective optimization problem or a constrained optimization problem, where some of the costs are used as constraints.

Nie and Li (2013) formulated an eco-routing problem as a mathematical program. The objective was to minimize the total travel costs, which were the monetary value of both fuel and time consumed from origin to destination. In addition, a constraint on CO₂ emission was imposed to the problem based on CO₂ emission standard. Different from previous research, this model was able considered more than one objective at the same time. Besides primary objectives such as fuel consumption and travel time, emissions (e.g. CO₂) were added as constraints and

the problem was formulated as a constrained shortest path problem.

Environmental related route choice problems with multiple objectives have also been studied in the context of traffic assignment problems. Tzeng and Chen (1993) proposed a traffic assignment model that simultaneously considered travel distance, travel time and CO emission in the objective function. CO emission on a link was modeled as a linear function of link traffic volume. Chen et al. (2011) formulated a traffic assignment model in such a way that CO emission was considered as a side constraint. Link CO emission was modeled as nonlinear function of link length and link travel time.

2.4 Traffic signal control and high-resolution traffic data

In this section, a brief overview of traffic signal operations will be provided. The overview is not intended as a complete description of traffic signal operations. Only these information related to this research will be discussed here. For detailed information, readers can refer to the traffic signal timing manual (Federal Highway Administration, 2008).

There are generally two types of traffic signal operation modes: pre-timed and actuated. Under pre-timed operation mode, the durations of signal phases are fixed in advance. This makes it relatively easy to incorporate the signal information into shortest path search algorithms, since whether a movement is allowed at a given intersection at a given time as well as the nearest available time window for that movement can be determined in advance. Thus, corresponding cost at the intersection can be calculated and used in the classical shortest path search algorithms such as label correcting algorithm or label setting algorithm

(Chen and Yang, 2000; Ahuja et al., 2002). It should be noted that delays caused by queues at the intersections are generally not considered in this paradigm.

More often, modern traffic signal controllers are designed in actuated operation mode. There are three types of operations in this category: semi-actuated, fully-actuated and coordinated. In semi-actuated mode, major movements are set to be default and operated as “non-actuated”. The movements from minor directions are detected and only permitted when necessary. In fully-actuated mode, all the phases are actuated. Cycle time is allocated based on the information detected from different approaches. There are generally no fixed cycle lengths when signal control operations are in semi-actuated or fully-actuated signal mode.

A more interesting signal control mode is coordinated actuated operation mode. In this mode, the cycle lengths at intersections are fixed and usually the same for the intersections in the same coordination zone. There are detections for all the approaches. For each phase, there are minimum green and maximum green time. The minimum green time defines the minimal green time that should be allocated to the phase and the maximum green time defines the maximum green time to the phase. The minimum green time is not effective for minor approaches as the phases for minor approaches may be skipped if no vehicles are detected during a cycle. For each phase, there is a force-off point, which defines the time within a cycle when the phase must end. And for each intersection, there is reference point which is used for the coordinations between intersections, whose time relationship is defined by offset. Offset defines the time relationship between coordinated phases at adjacent intersections. Different controllers (e.g. NEMA TS1, NEMA TS2, and the Type 170) have different choices of reference point. For the purpose of analysis in the shortest path problem, the reference point is assumed to be the start of coordinated phase in this research.

This research will be using high-resolution traffic data, including signal status data and loop detector data. By high-resolution, it means every loop detector actuation and every signal phase change will be detected and recorded as an event in the data collection system. From the vehicle-detector actuation data, headways and gaps between two consecutive vehicles can be easily derived. This sort of data can be obtained from data collection systems such as the SMART-Signal system developed by the University of Minnesota (Liu and Ma, 2009; Liu et al., 2009). It has been shown that queue profiles can be estimated using high-resolution traffic data (Liu et al., 2009).

Chapter 3

Stochastic Vehicle Routing Problems Considering Traffic Signals

In this chapter, we will study the vehicle routing problems considering traffic signals. It first specifies the problems of interest. Then, the Markov decision process (MDP) is introduced. Following this is the core part of this chapter, which shows how to formulate a path search problem considering traffic signals based MDP formulation. The basic formulation assumes the availability of historical high-resolution traffic data, but does not consider the situation where signal status information is available at real-time. Section 3.4 extends the basic formulation so that real-time signal status information can be utilized when available. Section 3.3.2 describes two different time discretization schemes and the conversion between them. Section 3.5 shows how the delay distributions used in the proposed model can be estimated from field data that are readily available.

3.1 Problem specifications

The problems studied in this chapter include:

1. a path search problem without real-time traffic signal information;
2. a path search problem with real-time traffic signal information.

Both of the two problems use expected travel time in their objective functions. The difference is whether real-time traffic signal information is used when searching for the optimal path.

The formulation of the first problem provides the fundamental mathematical tools. The formulation is based on Markov decision process (MDP) and requires the knowledge of the network topology and the configurations of the signal controllers. The structure of the network can be represented by a directed graph, which consists of nodes and directed arcs. The signal controller configurations include the types of signal controller, such as fixed timing, actuated, or coordinated signal control, the cycle lengths at each intersections, and offset settings in the case of coordinated signal control. It also requires the knowledge of intersection delay distributions and link travel time.

In the basic formulation, we assume the availability of historical high-resolution traffic data. But we don't assume the availability of real-time signal status information. The historical information needed by the algorithm can be obtained before the start of the trip. The algorithm, however, does require the vehicle's ability to know current time and its own position. The outcome of the proposed method will be a set instructions to choose turning directions based on vehicle's current location and current time.

If real-time traffic signal status information is available, we can extend the basic formulation so that the additional information can be utilized. And because

this will require the use of traffic signal status information at every intersection, a way to predict traffic signal status at downstream intersections is needed. This will help to reduce the uncertainty in our model.

3.2 Markov decision process (MDP)

A Markov decision process (MDP) is a discrete time stochastic control process. The system is described by a set of states S . At a given time step or stage $k \geq 0$, the system is in a state $s \in S$. There is an initial state s_0 at stage $k = 0$. At each stage k , the controller or decision maker chooses an action $a \in A_s \subseteq A$, where A_s is the set of available actions in state s and A is the action set. The system then randomly transits to a new state s' at next stage with probability $p(s'|s, a)$. The cost corresponding to this transition is a function of state s and action a and written as $c(s, a)$. So an MDP can be defined as a quadruplet,

$$\mathcal{M} = (S, \{A_s\}, c(s, a), p(s'|s, a)). \quad (3.1)$$

Please note the transition probability only depends on the current state s and action a , but not previous states and actions. This is a reasonable assumption in the context of shortest path problem in a traffic network, as how you can get to the destination only depends on where are you at this moment, but neither the origin nor the trip start time. The state and action at stage $k \geq 0$ are denoted by s_k and a_k respectively. The system's behavior, when k goes to infinity, is then described by a stochastic process $\{(s_k, a_k)\}_{k=0}^{\infty}$.

The mathematical model introduced above is very general. We usually impose some assumptions on the model. We assume that S and A_s do not vary with k . In this research, we also assume both S and A_s are finite sets. This assumption

eliminates many subtle mathematical issues when they are not discrete, but is sufficient for our application. In addition, we assume that $\sum_{s' \in S} p(s'|s, a) = 1$, which basically says there is no leakage out of the system during the process. Not requiring this assumption allows wider application of the model, but that is not the focus of this research. In the formulation, we consider the stage cost as a function of current state and action, $c(s, a)$. When the cost will also depend on the state at the next stage, we denote the corresponding cost by $c(s, a, s')$ and calculate the expected cost at current stage by

$$c(s, a) = \sum_{s' \in S} c(s, a, s') p(s'|s, a) \quad (3.2)$$

A decision rule, denoted by $d(s)$, is a procedure for action selection at a given state s . If the action is chosen with certainty, then we say it is a deterministic or pure. A randomized decision rule specifies a probability distribution on a set of actions. A decision rule is said to be Markovian if it only depends current state and action. It is said to be history dependent if it depends on past history of the system, i.e. the previous states and actions in addition to current state and action.

A policy, denoted by π , is a sequence of decision rules: $\pi = (d_0, d_1, \dots, d_k, \dots)$. When $d_k = d$ for all k , the policy is called stationary. In addition, if the decision rules in a policy are all Markovian, we call it a stationary Markov policy. It has been shown that a Markov policy is as good as history dependent policy (Puterman, 1994), which is the consequence of Markovian transition probability. When all the decision rules in a policy is deterministic, we call it a deterministic policy or pure policy. And we will focus on stationary deterministic policy most of the time in this research.

Each Markov decision process can be associated with a optimality criterion.

Depending on specific problems, the objective can be the optimized total cost or average cost. As the cost is also a random variable, we usually work on its expectation. In our problem, we are more concerned about the total cost, which can be defined as

$$u^\pi(s_0) = \lim_{K \rightarrow \infty} E\left\{\sum_{k=0}^{K-1} c(s_k, a_k) | s_0\right\} \quad (3.3)$$

, where s_0 is the initial state.

3.3 Model formulation

In this section, a path search problem will be formulated based on MPD with expected total cost as optimality criterion. Time and space information will be used to construct the state space. Time information comes from signal control parameters and the space information comes from the geometry of the underlying traffic network. The traffic network structure also decides the available actions at each state. The transition probability is constructed using delay distributions that can be estimated from field data. The stage cost will depend on current state and action.

3.3.1 State space and action

State space S is a set of mathematical objects that are used to describe the problem. It should contain the information that is essential to the problem. In this research, we focus on the shortest path problem considering traffic signals, which means the delays of a vehicle at intersections become the most important factor in the problem. The state space model should contain enough information to determine the delays of a vehicle at intersections.

Delays of a vehicle at an intersection largely depend on two things: the direction from which a vehicle approaches an intersection and the time when the vehicle arrives at the intersection. The approaching direction of a vehicle to an intersection is important because the green time is allocated to different approaches sequentially within a control cycle at the intersection. If two vehicles arrive at an intersection at the same time but from different directions, the delays they expect to experience are often different even if they both head to the same direction. For example, one vehicle arrives at an intersection from north and another vehicle arrives at the same intersection from west at the same time. Both of them are heading for the east, their delays (waiting time at the intersection) will be different because of traffic signal controls at the intersection. This makes it necessary for the state space model to include the immediate upstream intersection from which a vehicle travels from and the current intersection where the vehicle is.

For the same reason, the time when a vehicle arrives at an intersection is also important. Because most traffic signals are operated at cycle-by-cycle basis, cyclic delay patterns are usually observed at signalized intersections. For example, vehicles tend to experience similar delays if they arrive at the same intersection from the same approach at the beginning of green light during the morning peak hours even if their arrival times are in two different control cycles. On the other hand, the delay at an intersection for a vehicle arriving at the beginning of red light is usually significantly larger than that for a vehicle arriving during the middle of green phase at the same intersection. Given the importance of the relative time within a cycle, we define the time within a control cycle as cycle time and include it into the state space of the model.

Based on the analysis above, a state in our model is denoted by $s = (u, v, t)$, where $s \in S; u, v \in N; t \in T_v \subseteq T$. Here, u is the upstream intersection; v is

the current intersection; t is the vehicle's arriving time at current intersection v and T_v is the possible arrival times at current intersection v in a cycle. Then, the process of a vehicle traveling in a traffic network from an intersection to its downstream intersection is modeled as a state transition, which is also called a stage in MDP.

An example of state transition is shown in Figure 3.1. In this example, a vehicle arrives at intersection v from intersection u at cycle time t of intersection t . Then the vehicle chooses to travel towards intersection w . Depending on the delays at intersection v and travel time on the link, the vehicle will arrives at intersection w at cycle time t' , t'' , or t''' . A stage is between the arrivals at the two intersections.

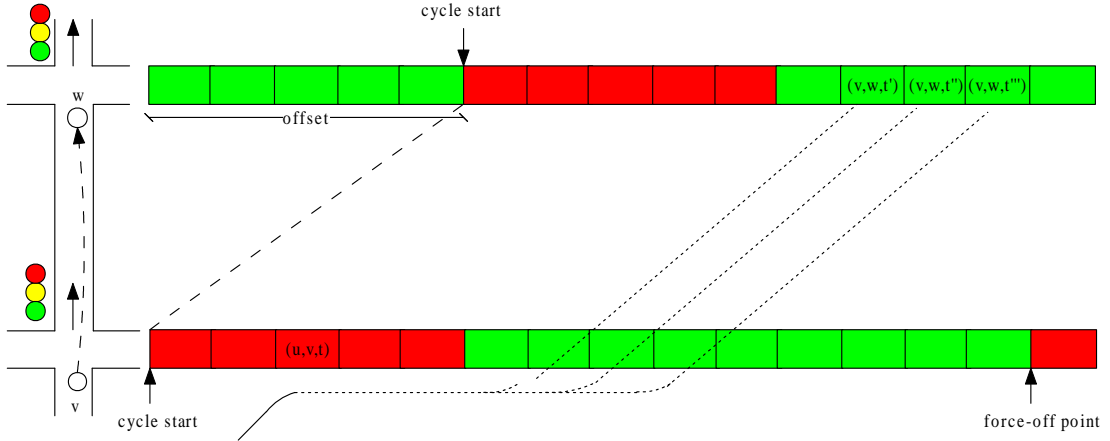


Figure 3.1: State transition at intersections with coordination

Time will be treated as discrete in this paper, as it is easier to model and discretization will be required for most numerical methods to solve the problem on a computer even if continuous time is used. But the granularity of time discretization can be different depends on requirements of specific applications.

A time discretization scheme with fine granularity, e.g. second-by-second possible arrival time, usually results in a model with high accuracy. At the same time, fine granularity means large size of the state space, as each possible arrival time needs to be represented by a state in the model. Consequently, the associated computational cost is expensive, as will be shown in Section 3.6 that elaborates the solution algorithm. Section 3.3.2 will talk more about time discretization scheme.

For each state $s = (u, v, t) \in S$, an action a is chosen from the set $A_s \subseteq A$. In a shortest path problem, this means to choose the next intersection to visit from the intersections connected by links emanating from v .

3.3.2 Time discretization

As we use discrete time in the model, it is easy to see that the size of the state space is dependent on the way we discretize the time, i.e. time discretization scheme. There can be various ways of time discretization. In general, a time discretization scheme with finer granularity has more descriptive power. On contrast, a more aggregated time discretization scheme usually lose some information when the aggregation process is carried out. What is the most appropriate level of aggregation really depends on specific applications, considering data availability, computational power, memory sources, and model accuracy, etc. Next, we will describe to time discretization scheme and how to convert from a time discretization scheme with finer granularity to a more aggregated time discretization scheme.

The basic time discretization scheme used in this paper is second-by-second time discretization scheme. In this time discretization scheme, each possible arrival second within a cycle at an intersection is represented by a state. More specific, for each intersection pair $(u, v) \in L$, the corresponding set of states is

$S_{(u,v)} = \{(u, v, t) : (u, v) \in L, t \in \{t_1, t_2, \dots, t_{\gamma(v)}\}\}$, where $\gamma(v)$ is the cycle length of intersection v . And the system state set becomes $S = \bigcup_{(u,v) \in L} S_{(u,v)}$. The State space model following this discretization scheme will be referred as “second-by-second state space model” in the following.

In the second time discretization scheme described in this dissertation, there are only two possible arrival times within a cycle at an intersection. Each of the possible arrival times is represented by a state. Thus, for each intersection pair $(u, v) \in L$, the corresponding set of states is $\tilde{S}_{(u,v)} = \{(u, v, \tilde{t}) : (u, v) \in L, \tilde{t} \in \{t_r, t_g\}\}$, where t_r and t_g represent the first time and second time period during a cycle respectively. And the system state space becomes $\tilde{S} = \bigcup_{(u,v) \in L} \tilde{S}_{(u,v)}$. State space model following this discretization scheme will be referred as “aggregated state space model” in the following. A side by side comparison of the two discretization scheme is shown in Figure 3.2.

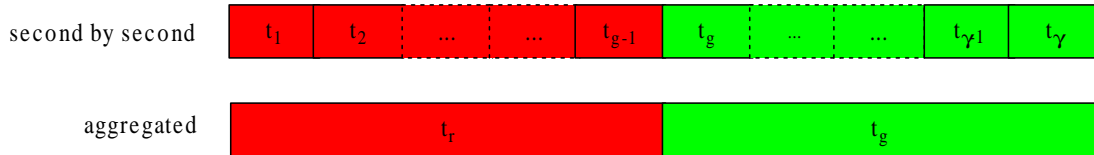


Figure 3.2: Two time discretization schemes

If we can collect second-by-second information, but don't have enough computational capacity, the time discretization scheme with finer granularity can be converted to a more aggregated one. The aggregation process reduces a process with large number of states to a new process with a smaller number of states. A requirement for the aggregation is that the original process is *lumpable* with respect to the aggregation. Lumpability ensures the new process is a Markov chain and the transition probabilities do not depend on the initial distribution (Kemeny

and Snell, 1976).

Suppose the original process is a Markov chain with state space

$$S = \{s_1, s_2, \dots, s_n\}.$$

The aggregation is denoted by $\tilde{S} = \{\tilde{s}_1, \tilde{s}_2, \dots, \tilde{s}_m\}$, where $m < n$. Each state \tilde{s}_i in \tilde{S} includes one or more state s_i in S and $\tilde{s}_i \cap \tilde{s}_j = \emptyset$. A necessary and sufficient condition for the Markov chain with state space S to be *lumpable* with respect to aggregation \tilde{S} is that, for every pair of \tilde{s}_i and \tilde{s}_j , the transition probability from a state $s_i \in \tilde{s}_i$ in the original model to the aggregated state \tilde{s}_j ($p_{s_i \tilde{s}_j}$) have the same value for every $s_i \in \tilde{s}_i$. The common values form the transition matrix of the lumped chain.

When the condition is satisfied, we first aggregate the state space. Then, we calculate the expected cost for each aggregated time interval by assuming uniformly distribution of arrival time during that interval in the finer time discretization scheme. Next, the transition probability is calculated accordingly by summing up all corresponding possibilities in the finer time discretization scheme. Finally, the possible actions stay the same for the states associated with a given intersection.

For example, we have a “second-by-second state space model”, whose state space is described as $S = \{(u, v, t) : (u, v) \in L, t \in \{t_1, t_2, \dots, t_{g(v)}, \dots, t_{\gamma(v)}\}\}$, where $\gamma(v)$ is the cycle length of intersection v and $g(v)$ is the green light start time of coordinated phase of intersection v . We want to convert this time scheme to the “aggregated state space model” we just introduced. As we have described in the last paragraph, the state space of the “aggregated state space model” will be $\tilde{S} = \{(u, v, \tilde{t}) : (u, v) \in L, \tilde{t} \in \{t_r, t_g\}\}$, which is constructed in the following

way

$$\tilde{s} = \begin{cases} (u, v, \tilde{t} = t_r) & \text{if } s = (u, v, t) : t \in \{t_1, t_2, \dots, t_{g(v)-1}\} \\ (u, v, \tilde{t} = t_g) & \text{if } s = (u, v, t) : t \in \{t_{g(v)}, t_{g(v)+1}, \dots, t_\gamma\} \end{cases} \quad (3.4)$$

The corresponding cost in the “aggregated state space model” is calculated as

$$\tilde{c}(\tilde{s}, a) = \begin{cases} E_t [c(s, a)] & \text{if } s = (u, v, t) : t \in \{t_1, t_2, \dots, t_{g(v)-1}\} \\ E_t [c(s, a)] & \text{if } s = (u, v, t) : t \in \{t_{g(v)}, t_{g(v)+1}, \dots, t_\gamma\}. \end{cases} \quad (3.5)$$

Finally, the transition probability is calculated by summing over all the associated transition probabilities in the “second-by-second state space model”. For example, the transition probability of a vehicle to transit from state $\tilde{s} = (u, v, \tilde{t} = t_r)$ to state $\tilde{s}' = (v, w, \tilde{t}' = t_r)$, i.e. a vehicle arrives at current intersection and downstream intersection both at the first arrival time, can be calculated as

$$\tilde{p}(\tilde{s}'|\tilde{s}, a) = \sum_{s'} p(s' \in \tilde{s}' | s \in \tilde{s}, a). \quad (3.6)$$

Since $p_{s\tilde{s}'}$ is the same for all the $s \in \tilde{s}$, so the calculation is only needed for one $s \in \tilde{s}$.

The following small example shows how the time discretization conversion is done in details. Suppose we arrive at intersection v from intersection u and choose to go intersection w . The cycle lengths at both intersection v and w are 4 seconds. Let's denote the state at intersection v by $s = (u, v, t)$ and that at intersection w by $s' = (v, w, t')$ in a “second-by-second state space model”. And we also denote the state at intersection v by $\tilde{s} = (u, v, \tilde{t})$ and that at intersection w by $\tilde{s}' = (v, w, \tilde{t}')$ in a “aggregated state space model”.

Based on our settings, t can be converted to \tilde{t} in the following way,

$$\tilde{t} = \begin{cases} 1, & \text{if } t \in \{1, 2\} \\ 2, & \text{if } t \in \{3, 4\} \end{cases}$$

The transition probability from state s to state s' in the “second-by-second state space model” and the cost at each possible state are given in Table 3.1.

Table 3.1: Second-by-second state transition probability and cost

$p(s' s, a)$		t'				$c(s, a)$
		1	2	3	4	
t	1	0.3	0.2	0.2	0.3	7
	2	0.2	0.3	0.4	0.1	5
	3	0.1	0.2	0.3	0.4	3
	4	0.2	0.1	0.4	0.3	3

Then, the corresponding costs and transition probabilities are calculated using Equation 3.5 and Equation 3.6. The results are given in Table 3.2. For example, $c(\tilde{s} = (u, v, \tilde{t} = 1), a)$ is calculated as

$$\begin{aligned} c((u, v, \tilde{t} = 1), a) &= E \left[c((u, v, t = 1), a) + c((u, v, t = 2), a) \right] \\ &= \frac{1}{2} \times 7 + \frac{1}{2} \times 5 \\ &= 6 \end{aligned}$$

And the transition probability $p((u, v, \tilde{t}' = 1) | (u, v, \tilde{t} = 1), a)$ is calculated as

$$\begin{aligned} p((u, v, \tilde{t}' = 1) | (u, v, \tilde{t} = 1), a) &= (0.3 + 0.2) = (0.4 + 0.1) \\ &= 0.5 \end{aligned}$$

Table 3.2: Aggregated state transition probability and cost

$p(\tilde{s}' \tilde{s}, a)$		\tilde{t}'		$c(\tilde{s}, a)$
		1	2	
\tilde{t}	1	0.5	0.5	6
	2	0.3	0.7	3

3.3.3 Cost function and transition probability

At each stage, vehicle travel time can be decomposed into two parts: 1) time for traversing the link, i.e. link travel time and 2) time for traversing the intersection, i.e. intersection delay. The delay at an intersection can be further decomposed into delay caused by traffic lights and delay caused by queuing vehicles in front of the current vehicle. Let δ_r denote the time difference between the arrival of the vehicle at intersection v and the next green start time for allowing it to travel to downstream intersection w , δ_q be the additional delay caused by the queuing vehicles in front, and τ denote the free flow travel time needed for the vehicle to traverse link (v, w) .

Delays at intersections, δ_q and δ_r , depend on many factors including queue lengths and red light durations, which vary from cycle to cycle, especially in the case of vehicle-actuated traffic signals. So it is more appropriate to model them as random variables. For free flow travel time (τ), it is more stable and can be either regarded as a constant or random variable for each link. As we can think of a constant as a special case of random variables, we considered τ as a random variable in the modeling part.

As the cost of interest is travel time, we want to calculate the stage travel time, i.e. the time cost between the arrivals of two consecutive intersections.

The travel time at each stage mainly depends the corresponding state and chosen action, so we write it as a function of the state $s = (u, v, t)$ and action $a(s) = w$, which is denoted by $c(s, a)$. And $c(s, a)$ is the expected waiting time at current intersection v plus the travel time from v to downstream intersection w given s and a , i.e. $c(s, a) = E[\delta_r + \delta_q + \tau | s, a]$, where $E[\cdot | \cdot]$ denotes the conditional expectation.

Note that

$$c(s, a) = 0 \tag{3.7}$$

where $s = \{(u, v, t) : v \in D, u \in D, t \in T_v\}$,

$$a = a(s) \in A_s,$$

The transition probability out of a destination set is given by

$$p(s' | s, a) = 0 \tag{3.8}$$

where $s = \{(u, v, t) : u \in N, v \in D, t \in T_v\}$,

$$s' = \{(v, w, t') : v \in D, w \notin D, t' \in T_w\},$$

$$a = a(s) \in A_s.$$

Equation 3.7 and Equation 3.8 mean the cost for staying in the destination set (D) is zero and once the system gets into there, it will not leave again. The destination set (D) is also called absorbing set in the literature.

The transition probability is the key factor to model the dynamics of a stochastic process. Suppose a vehicle arrives at intersection v at time t , and choose to visit intersection w . In other word, the system is in state $s = (u, v, t)$ at current stage and action $a(s) = w$ is taken. And the vehicle arrives at intersection w at

time t' at next stage, the system is in state $s' = (v, w, t')$. The corresponding transition probability, denoted by $p(s'|s, a)$, can be calculated as

$$p(s'|s, a) = p(t' = f(\delta_r + \delta_q + \tau)|s, a). \quad (3.9)$$

The specification of $f(\delta_r + \delta_q + \tau)$ depends the specific state space model used. In the following, the specification of $p(s'|s, a)$ will be described assuming *second-by-second state space model* is used.

Different situations need to be considered when specifying transition probability. First, let's consider the situation where intersections v and w are coordinated with offset ξ and background cycle length γ . Then the transition probability can be calculated as

$$p(s'|s, a) = p(t' = (t + \delta_r + \delta_q + \tau - \xi) \% \gamma | s, a) \quad (3.10)$$

, where $\%$ is the modulo operation.

Note only δ_r , δ_q , and τ are random variables in this formula. So the essential information needed to construct the transition probability is $p(\delta_r + \delta_q + \tau | s, a)$, i.e. the distribution of the sum of the intersection delay ($\delta_r + \delta_q$) and travel time (τ) along the link.

The principle behind Equation 3.10 can be explained by Figure 3.3. Because intersections v and w are coordinated, they will have the same background cycle length γ with an offset ξ . Consequently, the possible arrival times (in terms of cycle time) at both intersections are in the set of $T = \{1, 2, \dots, \gamma\}$. And when a vehicle arrives at v at $t \in T$, the corresponding cycle time at w should be $\tilde{t} = (t - \xi) \% \gamma$. Given the intersection delay at v is $\delta_r + \delta_q$ and travel time on link (v, w) is τ , the vehicle arrives at w at cycle time $t' = (\tilde{t} + \delta_r + \delta_q + \tau) \% \gamma = (t + \delta_r + \delta_q + \tau - \xi) \% \gamma$.

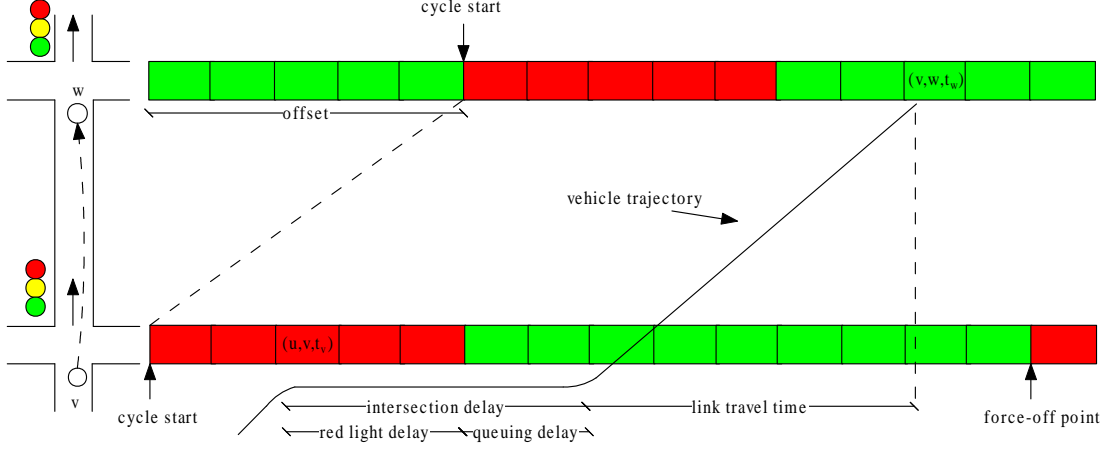


Figure 3.3: Transition probability at intersections with coordination

For cases where adjacent intersections are not coordinated, it is assumed the possibility of arrival times at downstream intersections is uniformly distributed, i.e.

$$p(s'|s, a) = \frac{1}{\gamma(w)} \quad (3.11)$$

, where $\gamma(w)$ is the cycle length at w .

If the system is in state $s = (u, v, t)$ at stage k and transits to state $s' = (v, w, t')$ at stage $k + 1$. The transition probability to state s' by choosing action a in state s at stage k is $p_k(s'|s, a)$. If the transition probability doesn't depend on stage k , the transition probability is called stationary and can be simply written as $p(s'|s, a)$. This study mostly focuses on the stationary transition probability. Note that different stationary transition probabilities can be used for different times of day, e.g. morning peak, afternoon peak, etc.

3.4 Incorporation of real-time traffic signals

With a state space model $s = (u, v, t)$, the algorithm only uses real-time information about vehicle locations and arrival times. But if real-time signal status information is also available and can be delivered to vehicles at real time, such information would help to provide more accurate prediction of transition probabilities and delays at intersections. To make use of real-time signal status information, we include signal status (r) into the state space model and it becomes,

$$s = (u, v, t, r), \quad (3.12)$$

where u is the upstream intersection, v is the current intersection, t is the arrival time at the current intersection, and r is the signal status. In the following of this subsection, let $s = (u, v, t, r)$ denotes the current state and $s' = (v, w, t', r')$ be the next state.

As we discussed earlier, the signal status at a given cycle time for an intersection is not deterministic for vehicle actuated traffic signals. However, when a vehicle arrives at a given intersection, the current signal status is known and such information can be used to improve the estimation of delay distributions. More concretely, when we estimate the delay distributions at current intersection without knowing current signal status, all the possible delays are included no matter the signal status is green or red. But when the current traffic signal status is known, only the delays corresponding to either green or red traffic light need to be used, so we are more certain about the delays.

For downstream intersections, it should be possible to estimate the probability of signal status at a given specific cycle time using high-resolution traffic data and corresponding signal parameters, as shown by Hu and Liu (2013). Let this

probability be $p(r'|(v, w, t'))$. Then, the state transition probability with real-time signal status information is calculated as

$$p((v, w, t', r') | (u, v, t, r), a) = p((v, w, t') | (u, v, t, r), a) p(r' | (v, w, t')) \quad (3.13)$$

, where $p((v, w, t') | (u, v, t, r), a)$ is calculated by Equation 3.10 or Equation 3.11.

3.5 Estimation of intersection delay distributions

As can be seen from the last section, both the calculations of stage costs and transition probabilities depend on distributions of intersection delays, which include red light delay distributions and queuing delay distributions conditioning on state-action pair, i.e. the distributions of random variables $\delta_r|(s, a)$ and $\delta_q|(s, a)$.

In this section, we will show that these distributions can be estimated using high resolution traffic data, which can be collected by systems like the SMART-Signal system developed by the University of Minnesota. By high resolution, it means every loop detector actuation and every signal phase change will be detected and recorded as an event in the data collection system. From the vehicle-detector actuation data, headways and gaps between two consecutive vehicles can be easily derived. A detailed description of the system can be found in (Liu and Ma, 2009; Liu et al., 2009).

It has been shown that queue profiles can be estimated using high resolution traffic data (Liu et al., 2009). Given queue profiles and signal status information, red light delays and queuing delays can be calculated. Next, we will give some examples of intersection delays, including red light delays and queuing delays, based on field data from TH-55 at west of Minneapolis, Minnesota, USA. These

data were obtained from an advanced detector at Rhode Island Ave. (detector number 10 in Figure 3.4) during morning peak hours in September, 2008.

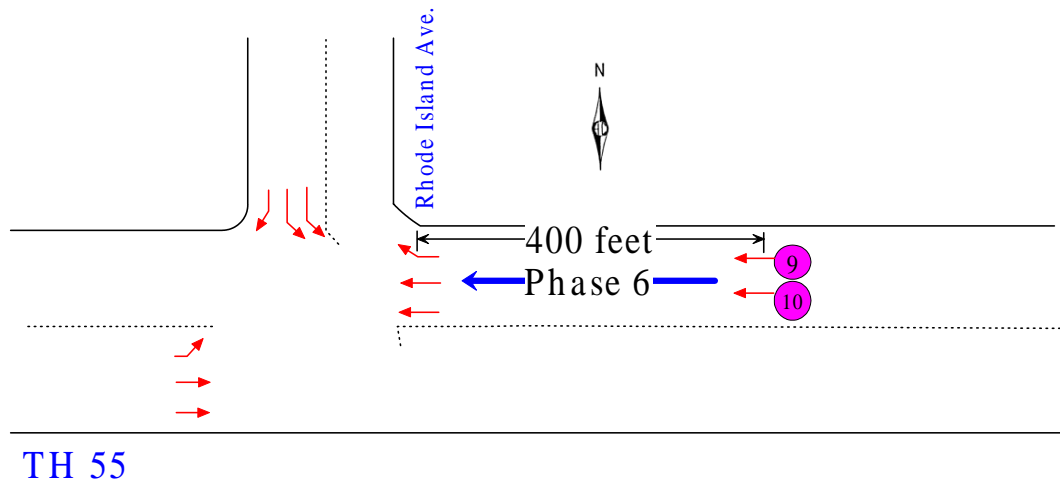


Figure 3.4: Sample data collection site

The queue profiles of two cycles during morning peak hours (September 2, 2008) are generated based on the method developed by Liu et al. (2009) and shown in Figure 3.5. The cycle information is given in Table 3.3. It can be seen that the red light durations and cycle lengths are the same for these two cycles. But their queue profiles are different. The maximum queue length of the first cycle is less than 400 *ft*, while the maximum queue length of the second cycle is more than 800 *ft*. This is no surprise as the traffic volumes vary from cycle to cycle.

Because of different queue profiles, the intersection delays experienced by a vehicle are different even if it approaches the intersection at the same cycle time. For example, when a vehicle approaches the intersection with a trajectory indicated by solid black line in Figure 3.5, the red light delay and queuing delay

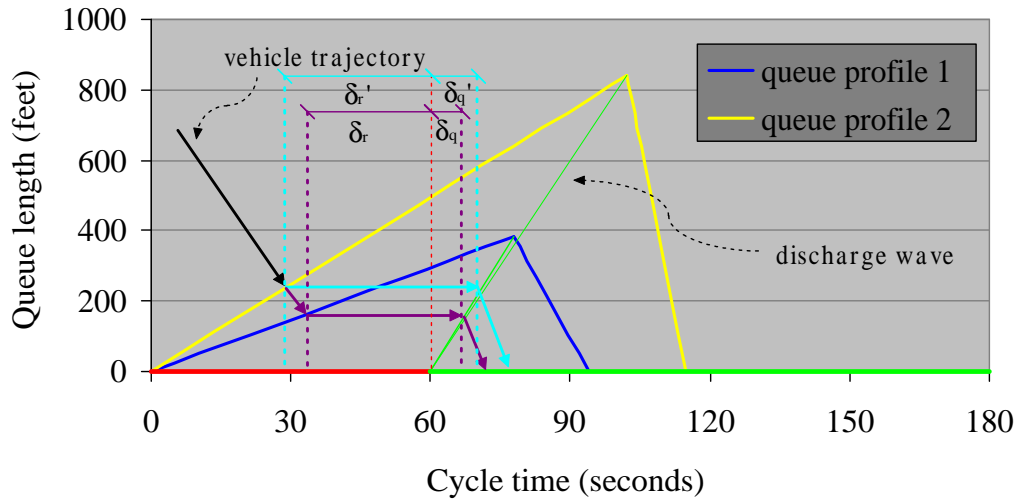


Figure 3.5: Intersection delays with different queue profiles

Table 3.3: Sample cycle information

	Red starts	Green Starts	Cycle ends	Cycle length	Red light time
Cycle 1	07:47:16	07:48:16	07:50:16	180 s	20 s
Cycle 2	07:50:16	07:51:16	07:53:16	180 s	20 s

corresponding to the first cycle (δ_r and δ_q) are different from those corresponding to the second cycle (δ'_r and δ'_q).

For vehicle-actuated traffic signals, the duration of red lights and green lights may also vary from cycle to cycle. For fully-actuated traffic signals, actual durations of green lights depend on time gaps between vehicle arrivals and signal control parameters such as minimum green, maximum green, and green extension. For coordinated phases at an intersection using semi-actuated traffic signals, green light durations may be increased when the green time allocated to minor

approaches is not fully used. This phenomenon is usually referred as “early return to green” in the literature. Given background cycle length is the same from cycle to cycle, the red light delays become different for the same arrival time at the intersection.

Provided the queue profile and signal status information, it is easy to calculate the red light delay and queuing delay for a vehicle conditioning on its arrival time and moving direction. When the queue can be discharged within the next available green time for the moving direction, the red light delay is just the time difference between the vehicle arrival time and green start time. The queuing delay is just the time needed to discharge the queue in front of the vehicle.

In over saturated situation, however, the queuing vehicle in front of the current vehicle cannot be fully discharged during the first available green time. Current vehicle has to wait for at least one more cycle. The green times before the cycle during which current vehicle can be discharged is unusable for it and should be regarded as red time. So the red light delay of current vehicle is calculated as the time difference between its arrival time at the intersection and start of green time of the cycle during which it can be discharged. The queuing delay is the time needed to discharge the queue in front of the vehicle at the start of green time at that cycle.

Based on the analysis above, we can generate samples of red light delays and/or queuing delays conditioning on arrival time and moving direction for each cycle where we have signal and/or queue information. Provided large amount of historical data, empirical distributions of red light delays and queuing delays can be generated.

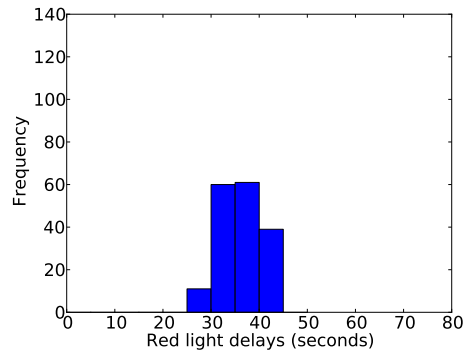
In Figure 3.6, we give some sample distributions for red light delays, queuing delays, and intersection delays. For a given time and moving direction, intersection

delay equals red light delay plus queuing delay. We still use the data from detector number 10 in Figure 3.4. And we use the data during morning peak hours (7:00 am - 9:00 am) of five workdays in a week (9/8/2008 - 9/12/2008). But please note that these delays in Figure 3.6 are not conditioned on signal status, so they are only good for the state space model without traffic signal information. But the estimation of delay distributions for the state space model with traffic signal information is straightforward following this. One just need to calculate delays conditioning on signal status and get the conditional distributions.

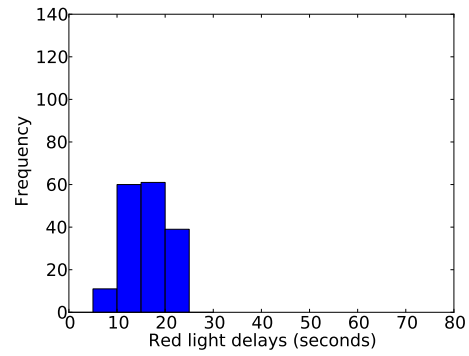
The background cycle length at this intersection is 180 seconds. We calculated the red light delays, queuing delays and intersection delays for phase 6 at cycle time 20 sec and 40 sec. In other words, the delay distributions are for a vehicle, arriving at the intersection cycle time 20 sec and 40 sec and traveling from the east and the west. The distributions corresponding to cycle time 20 sec are shown in the first column in Figure 3.6 and those corresponding to cycle time 40 sec are shown in the second column.

From Figure 3.6(a) and Figure 3.6(b), it can be observed that the red light delays of cycle time 20 sec are generally longer than those of cycle time 40 sec. This is simply because a vehicle arriving earlier at an intersection usually waits longer for the green start. Red light delays become small when some phases for minor approaches are skipped because no vehicles are detected for those minor phases. On the other hand, the queuing delays for early arrival vehicles are generally shorter, which can be seen from Figure 3.6(c) and Figure 3.6(d). This is due to the fact that the queue in front of a vehicle is usually shorter if it arrives earlier within a cycle, assuming green end to be the start of a cycle.

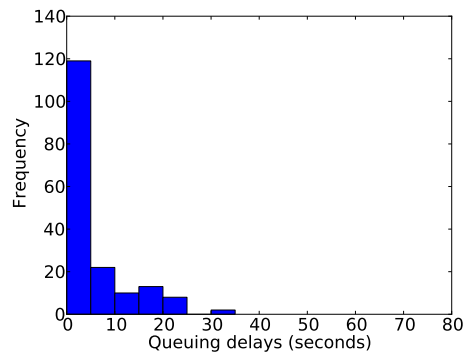
By combining red light delays and queuing delays together, we get the distributions of intersection delays. The shapes of intersection delays become somewhat



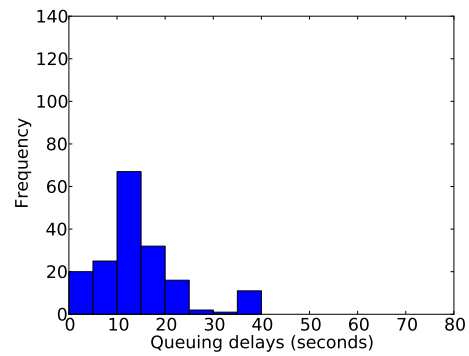
(a) Red light delays at cycle time 20 sec



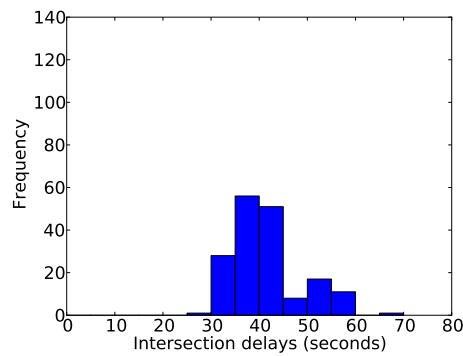
(b) Red light delays at cycle time 40 sec



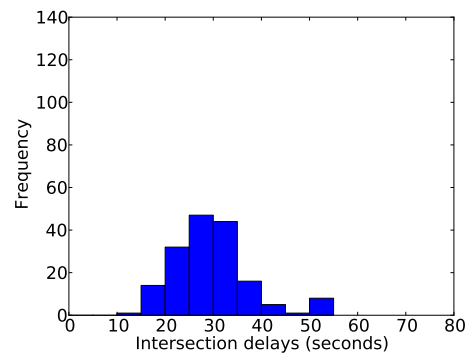
(c) Queueing delays at cycle time 20 sec



(d) Queueing delays at cycle time 40 sec



(e) Intersection delays at cycle time 20 sec



(f) Intersection delays at cycle time 40 sec

Figure 3.6: Examples of delay distributions

similar for these different arrival times. There is usually a trade-off between red light delays and queuing delays for vehicles experiencing delays at intersections. Long red light delays usually correspond to short queuing delays, and vice versa.

3.6 Value iteration method

Given the optimal policy exists, which is always the case for MDP with finite state space and action space, it can certainly be found by brute force algorithm, that is to go over all policies. But there are $|A|^{|S|}$ policies in total, where $|A|$ and $|S|$ are the cardinality of set A and S , so it is too slow when the problem size is reasonably large.

Instead of focusing on the policies, it is also possible to search over values of states and compute the resulting policy. For stochastic shortest path problem with any initial conditions \mathbf{u}_0 , it has been shown that the sequence $u_m(s)$ generated by the following dynamic programming iteration (Bertsekas (1995), chapter 7.2 of volume I),

$$u_{m+1}(s) = \min_{a \in A_s} E_{s'} [c(s, a) + u_m(s')], \quad (3.14)$$

where m is the index of the dynamic program, converges to $u^*(s)$ for each $s \in S$ and $u^*(s)$ satisfy the *Bellman optimality equation*

$$u^*(s) = \min_{a \in A_s} E_{s'} [c(s, a) + u^*(s')]. \quad (3.15)$$

Two assumptions are required for the existence and uniqueness of the solution to Equation 3.15. In the context of the shortest path problem, it simply means 1), there exists at least one policy with which the target states will be reached with positive probability after finite number of stages (existence of *proper policy*); 2),

the cost $c(s, a)$ for non-destination states are strictly positive. These assumptions are satisfied in SPP with traffic signals. So the MDP formulated in earlier this chapter can be solved by value iteration method. One possible version of pseudo code for the MDP algorithm is given in Algorithm 1.

Algorithm 1: Value iteration algorithm for infinite horizon MDP

Data: traffic network, signal settings, intersection delay distributions, link travel time distributions, destination (D)

Result: expected time cost at each state ($u(s)$), the navigation policy to the destination with minimal expected time cost ($\pi^*(s)$)

for *each state* $s \notin D$ **do**

$u_0(s) = \infty$;

end

for *each state* $s \in D$ **do**

$u_0(s) = 0$;

end

while $\| \mathbf{u}_m - \mathbf{u}_{m-1} \|_\infty > \varepsilon, \varepsilon > 0$ **do**

$m = m + 1$;

for *each state* s **do**

for *each action* a **do**

 compute $Q_m(s, a) = c(s, a) + \sum_{s'} p(s'|s, a) u_{m-1}(s')$;

end

 compute and store $\pi_m^*(s) = \underset{a}{\operatorname{argmin}} Q_m(s, a)$;

 compute and store $u_m^*(s) = Q_m(s, \pi_m^*(s))$;

end

end

return $\pi_m^*(s), u_m^*(s)$

The number of iterations required to achieve a given accuracy, i.e. the difference between current values and the optimal values, can be bounded given the ratio $\| \mathbf{u}^* \|_\infty / \underline{c}$ is known, where $\| \mathbf{u}^* \|_\infty$ is maximum norm of the optimal objective value vector and $\underline{c} = \min_{s \in S \setminus D, a \in A_s} c(s, a)$ (Bonet, 2007). During each iteration, the inner for loop of the algorithm goes over all the actions available to a state and the corresponding successor states, which requires $O(|S||A|)$ steps. And the number of steps required for each value iteration is $O(|S|^2|A|)$, as there are $|S|$ state updates in each iteration. In our problem, $|A|$ is usually small, as there are only limited number of turning choices available to a vehicle at an intersection, usually 2 or 3. The size of set S depends on the size of the network N and the number of possible arrival times at intersections, i.e. $|S| = \sum_{(v,w) \in L} |T_{(v,w)}|$.

Given a specific network, the size of N is fixed. In this situation, the size of set S only depends the number of arrival times, which is decided by time discretization scheme. A time discretization scheme with fine granularity results in a large number of possible arrival times. This increases the computational burden, but is likely to increase the accuracy of the model as well.

The algorithm stops when the value difference between the consecutive iterations $\| \mathbf{u}_m - \mathbf{u}_{m-1} \|_\infty$ is smaller than a threshold ε . The difference between the optimal value and value of current iteration $\| \mathbf{u}^* - \mathbf{u}_m \|_\infty$ can be bounded by a function of ε (Hansen, 2011). This implies that the computational cost of the algorithm depends on the choice of the parameter ε . When the computational resources are limited, one can adjust ε to get the solution within reasonable amount of time.

Chapter 4

Stochastic Eco-routing Problems Considering Traffic Signals

In Chapter 3, we have solved vehicle routing problems that minimize expected total travel time. Besides travel time, people may have other costs of concern when they travel. For example, one may be more concerned about the fuel efficiency, in which case it makes more sense to minimize the fuel consumption instead of travel time. Recently, people have formulated the vehicle routing problems with environmental costs in consideration as “eco-routing” problem. To solve this type of problems, it requires efforts to calculate environmental related costs. In this chapter, we will solve the eco-routing problem explicitly considering traffic signals. In addition, we will also study the problem with multiple costs of interest based on the constrained formulation of MDP.

4.1 Problem specifications

The problems studied in this chapter include:

1. an eco-routing problem;
2. an eco-routing problem with constraints.

An eco-routing problem is essentially a path search problem. Instead of travel time, environmental costs are considered in an eco-routing problem. As the inherent uncertainty from vehicle-actuated traffic signals remains there, we are still facing a stochastic vehicle routing problem. In addition, extra efforts are needed in the calculation of environmental costs. This involves the integration of microscopic vehicle emission models and how to obtain necessary inputs for the models.

Furthermore, we will consider multiple costs of interest in the problem, as there are usually more than one environmental costs related to vehicle activities. We will transform our formulation into a linear programming formulation, which allows us to add constraints with regards to different costs. In a constrained formulation, one of the cost will be used as the primary objective and other costs will be considered in the constraints. In this way, we are able to simultaneously consider several costs of interest.

4.2 Model formulation

We will still build our model based on MPD. The state space needs to include useful information about eco-routing problem. As can be seen later, the shape of vehicle trajectory is the crucial input in the estimation of vehicle emissions and fuel consumption. The vehicle trajectory on a link largely depend on whether the vehicle stops at the upstream intersection as well as the downstream intersection. And the stops, in turn, depend on a vehicle's arrival times at both intersections. So a good choice of the state space should include the information at current

intersection (v) , down stream intersection (w) , and a vehicle's arrival times at both intersections t_v, t_w . Mathematically speaking, the state space of a stochastic eco-routing problem can be written as,

$$S = \{(u, v, w, t_v, t_w) : u \in U_v, v \in N, w \in W_v, t_v \in T_v, t_w \in T_w, T_v \subset T, T_w \subset T\}$$

, where U_v is the set of upstream intersections of intersection v ; N is the set of intersections in the network; W_v is the set of downstream intersections of intersection v , and T is the set of vehicle arrival times at intersections.

Compared this with the state space model in Chapter 3, it is more complex. The size of the state space will be increased by a factor of $N \times T_w$. This number becomes significant when N is large. To address this issue, we introduce a reduced state space model in the following.

As can be shown later, it is not necessary to include the information from the downstream intersection with mild assumptions. This can simplify our state space to

$$S = \{(u, v, t_v) : u \in U_v, v \in N, t_v \in T_v, T_v \subset T\} \quad (4.1)$$

This state space model is consistent with the one we used in Chapter 3 and can reduce the computational burden compared to the one with information at both ends of links. So we will use the state space model specified by Equation 4.1 in the following.

With state space model specified, we can calculate the transition probability $p(s'|s, a)$ following the same method as described in Section 3.3.3 of Chapter 3. The available actions at each state are also the set of immediate downstream intersections accessible from current state.

When the cost of interest is environmentally related, the stage cost is the environmental cost that corresponds to the vehicle activity at one stage. This includes the environmental cost results from the vehicle movement on a link and the stop at an intersection. As will be shown shortly in Section 4.3, this stage cost mainly depends the vehicle trajectory at each stage. And the vehicle trajectory is affected by the value of intersection delays, so the stage cost is also a function of intersection delays. From Chapter 3, we can see that intersection delays can be expressed as a function of current state and action. So we may also be able to write the environmental stage cost as a function of current state and action, denoted by $c(s, a)$. But the calculation of environmental cost is so complicated that we can not express it in an closed form as we do for travel time. The details of obtaining $c(s, a)$ for environmental costs will be given next.

4.3 Estimation of eco-costs

In an eco-routing problem, the costs of interest are vehicle emissions and/or fuel consumption. They depend on many factors including vehicle characteristics, road characteristics, and traffic conditions, etc. All the information is used as input to a microscopic vehicle emission model.

Some of the information, such as vehicle characteristics (e.g. vehicle make, year) and road characteristics (e.g. grade), is static. So it is relatively easy to prepare these information for a vehicle emission model. Some of the other information, such as relative humidity and temperature, is dynamic, but they are mostly independent of traffic conditions. We assume all the information is available as inputs to a vehicle emission model.

Another important input to a vehicle emission model is second by second speed,

i.e vehicle trajectory. This is the most traffic related input to a vehicle emission model. Because of uncertainty in traffic condition, it is challenging to obtain such information for path search problem. This is especially true for signalized network because of the disruption to traffic from traffic signal controls. Next, we will introduce a method to estimate vehicle trajectories based on the information of traffic signal status and vehicle arrivals at intersections.

To estimate the vehicle trajectories in a signalized traffic network, we first make some assumptions about vehicle's behaviors at intersections and on links. We first assume a vehicle either stops at an intersection or pass by the intersection with free flow speed. This is also the approach suggested by EPA for analyzing carbon monoxide of intersection project (EPA, 2010). When a vehicle starts to move from a stop, we assume it always accelerates from zero speed to free flow speed with constant acceleration rate. And when a vehicle decelerates, we assume it always make a full stop with constant deceleration rate.

With these assumptions, a vehicle at an intersection can be in one of the two status: stop or travel at free flow speed; and a vehicle on a link can be in one of three status: deceleration, travel at free flow speed, or acceleration. When a vehicle stops or travels at free flow speed, its acceleration rate is zero. When a vehicle accelerates or decelerates, it changes its speed with constant acceleration rate or deceleration rate. Figure 4.1 gives an overview of vehicle status transitions at intersections and on links.

One more assumption is needed before we proceed to calculate vehicle trajectories on links. When a link length is small and a vehicle stops at both ends of the link, it is possible that the vehicle needs to decelerate before it accelerates to its desire speed. But this situation may not happen so often, as short links usually appear in urban street network where speed limit is quite low. Consequently, it

is reasonable to assume that links are long enough for vehicles to accelerate from zero speed to desire speed and then decelerate to zero speed. If this assumption is violated, we may over estimate the environmental costs on short links, which will be discussed later.

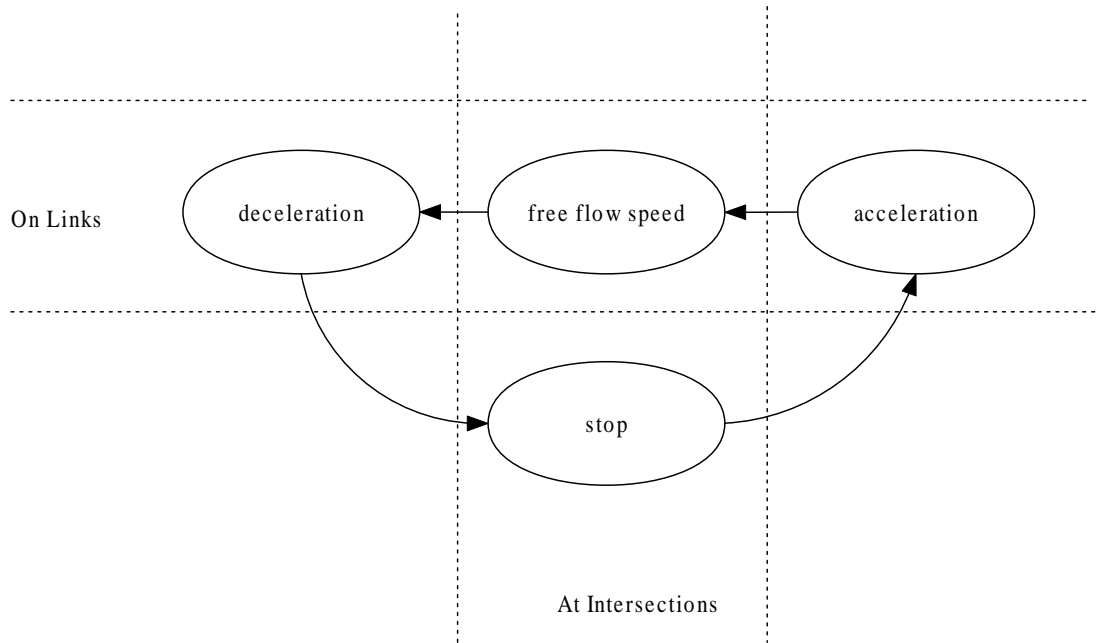


Figure 4.1: Vehicle status transitions at intersections and on links

Depending on a vehicle status at upstream and downstream intersections of a link, the trajectory of the vehicle on the link can be one of the following type (Figure 4.2):

- Type I: vehicle stops at both intersections;
- Type II: vehicle stops at upstream intersection;
- Type III: vehicle stops at downstream intersection;

- Type IV: vehicle stops at neither intersection.

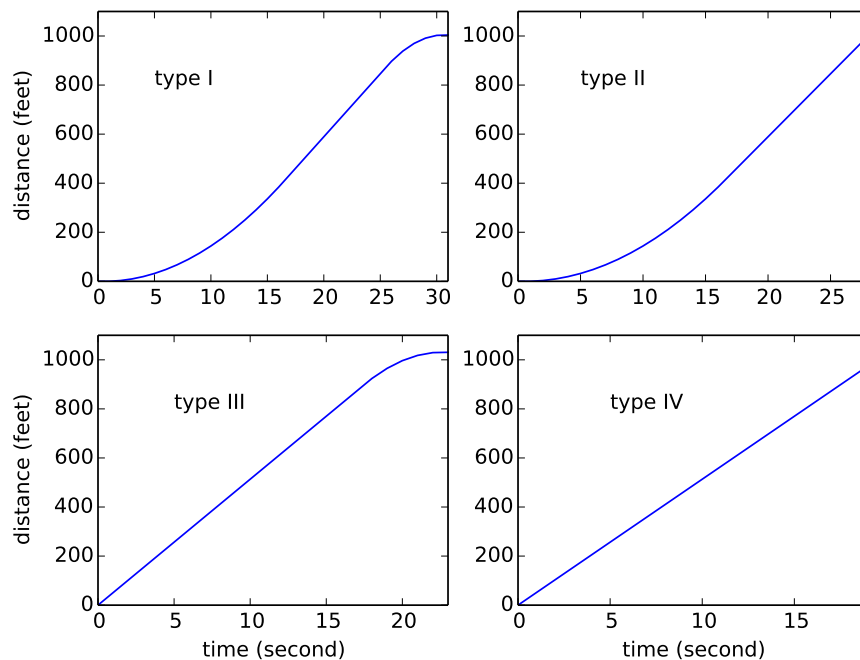


Figure 4.2: Vehicle trajectory types on links

Once trajectory type is determined, vehicle trajectory can be estimated given link length, acceleration rate, deceleration rate, and desire speed. So the vehicle trajectory on a link depend on the vehicle status at both ends of the intersections.

It is obvious that the trajectories can be different just because of the link length, even if everything else is the same. Figure 4.3 shows two different trajectories with same free flow speed, acceleration rate, and deceleration rate. And the vehicle stops at both ends of the link. The only difference is the link length. But it should also be noted that the two trajectories shown in Figure 4.3 are fundamentally the same. They only differ by the portions covered by the vehicle traveling at

free flow speed. The acceleration portions and the deceleration portions of these two trajectories stay the same, respectively.

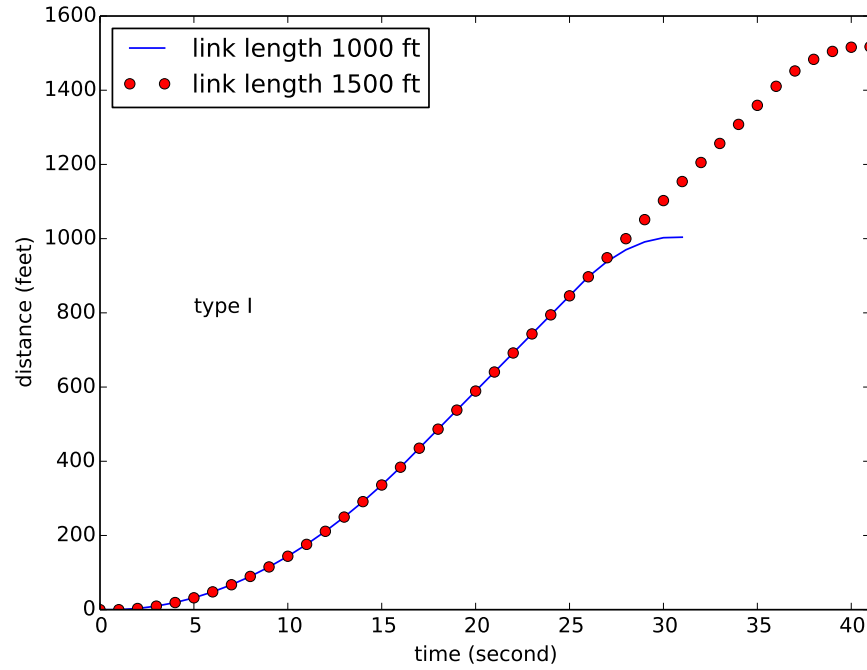


Figure 4.3: Vehicle trajectories on links with different lengths

To better understand a trajectory on a link, we divide a link into three parts according to trajectory type I, assuming acceleration rate, deceleration rate, and free flow speed are fixed. As shown in Figure 4.4, the first part of the link (x_1) corresponds to the acceleration process. It starts from the upstream intersection of the link and the length of the first part equals to the distance needed by a vehicle to accelerate from a full stop to free flow speed. The third part of the link (x_3) corresponds to the deceleration process. Its length equals to the distance needed by a vehicle to decelerate from free flow speed to a full stop. And it ends

at the downstream intersection of the link. The remaining part of the link is the second part (x_2), on which the vehicle travels at free flow speed.

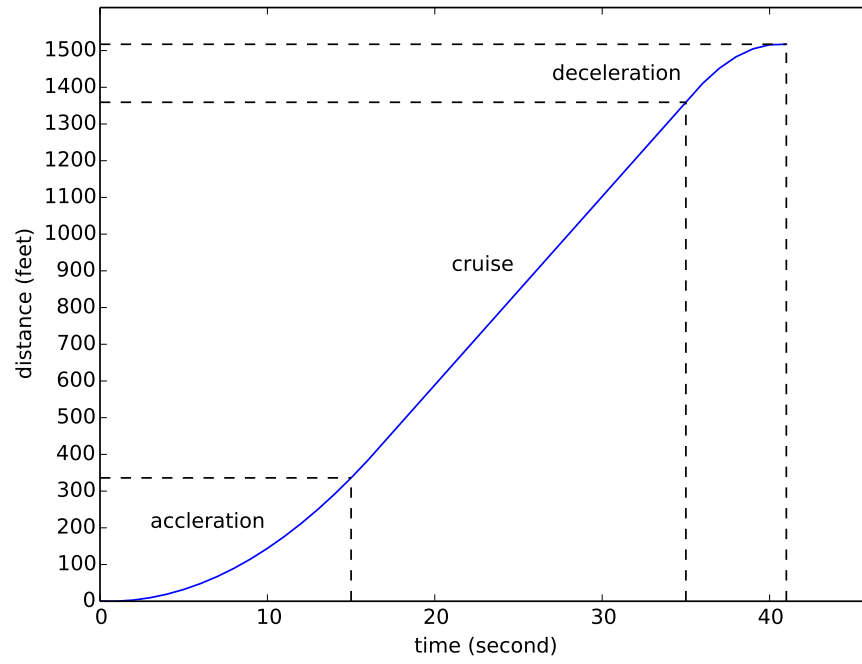


Figure 4.4: A vehicle trajectory on a link

Let's denote the length of a given link by x , free flow speed on this link by \dot{x} , acceleration rate on this link by \ddot{x}_a , and deceleration rate on this link by \ddot{x}_d . Simple physics allows us to determine the values of x_1, x_2, x_3 by the following equations:

$$\begin{aligned}
x_1 &= \frac{\dot{x}^2}{2\ddot{x}_a} \\
x_2 &= x - x_1 - x_3 \\
x_3 &= \frac{\dot{x}^2}{2\ddot{x}_d}
\end{aligned} \tag{4.2}$$

As the values of x , \dot{x} , \ddot{x}_a , and \ddot{x}_d are link specific, so do the values of x_1, x_2, x_3 .

The benefit of dividing a link trajectory into three parts is that we can disentangle the vehicle status at upstream intersection of a link from that at the downstream intersection of the same link when deciding the link trajectory type. As can be seen from the analysis above, the vehicle trajectory type is decided by the vehicle status at both ends of a link. But we only have the information for current intersection in our state space model. Although it is possible to expand the state space model so that information at both intersections can be included, this approach will significantly increase the size of the state space, and thus the computational burden. Instead, we introduce another approximated approach based on link length division, so that we can achieve the goal without the need of increasing state space size.

One technique to integrate emission model with transportation model is to use velocity/acceleration-indexed lookup tables. But this eliminates the time dependence in vehicle emissions on vehicle operation history, which can be significant to instantaneous emission values (Frey et al., 2001). To be as accurate as possible, the vehicle trajectories are used as inputs to the microscopic emission models in our method. But at the same time, some compromises are made in consideration to the feasibility and efficiency. Specifically, the complete trajectory along a route is divided into pieces by links. Although correlations between link trajectories are

modeled, they are put into microscopic model one by one and the time dependence in vehicle emissions between link trajectories is ignored. Furthermore, one link trajectories are divided into three parts, between which the time dependence in vehicle emissions is also ignored.

In our formulation, one step cost includes the cost at current intersection and the cost on immediate downstream link. The environmental costs at intersections are generated when a vehicle is idling. When traveling on a link, a vehicle can be in one of the three status on a link: acceleration, deceleration, and free flow speed travel, because of our assumptions. Considering this together with the link division mentioned earlier, we define the following variables,

- Link cost vector $\mathbf{C}_k^{gh} \in \mathbb{R}^b$ represents vehicle emission and fuel consumption costs for a vehicle traveling on link part g when it is in status h at stage k . The values of $h \in \{1, 2, 3\}$ correspond to acceleration, free flow speed travel, and deceleration. Superscript b denotes the number of environmental costs of interest.
- Intersection cost vector \mathbf{I}_k is a vector denoting the expected vehicle emissions and fuel consumption when idling at stage k . It is a function of idling emission/fuel consumption rate and idling duration.

Because we have divided a link into three parts, the shapes of the vehicle trajectory on different parts of the link do not affect each others. And by our assumption, the environmental costs related to different parts are also independent from each others. Since we don't have information at downstream intersection at current state, we assume a vehicle always travels on the third part of the link at free flow speed. When we calculate the costs at the immediate downstream

intersection, corrections to the costs are made accordingly, which will be described next.

As we assume a vehicle either pass by an intersection with cruise speed or stop at an intersection, the vehicle needs to accelerate from zero speed on the downstream link if it stops at current intersection. Also because we have assumed the link is long enough and vehicles always accelerate to desire speed from zero speed with constant acceleration rate, the trajectory for the acceleration process stays independent of the links. Consequently, \mathbf{C}^{33} and \mathbf{C}^{32} , as well as their difference $\Delta\mathbf{C}$, stay the same for each link.

When calculating the stage costs, we always assume a vehicle travels at free flow speed on the last part of a link and the corresponding costs are given by \mathbf{C}^{32} . In the case where the vehicle stops at current intersection, the difference in costs of last step will be corrected by adding back the difference ($\Delta\mathbf{C}$) of the upstream link. More precisely, the costs at stage k can be written as

$$\begin{cases} \mathbf{c}_0 &= \mathbf{C}_k^{12} + \mathbf{C}_k^{22} + \mathbf{C}_k^{32} & \text{if not stop} \\ \mathbf{c}_1 &= \mathbf{C}_k^{11} + \mathbf{C}_k^{22} + \mathbf{C}_k^{32} - \mathbf{C}_{k-1}^{32} + \mathbf{C}_{k-1}^{33} + \mathbf{I}_k & \text{if stop} \end{cases} \quad (4.3)$$

And one step cost $c(s, a)$, i.e. the step cost when action a is taken in state s , becomes

$$\mathbf{c}(s, a) = \mathbf{c}_0 \times p(\delta = 0|s, a) + \sum_{\delta \neq 0} \mathbf{c}_1(\delta) \times p(\delta|s, a) \quad (4.4)$$

, where $\delta = \delta_r + \delta_q$ is the possible delays at current intersection.

As we discussed earlier, when a link is short and a vehicle stops at both ends of the link, the vehicle may have to decelerate before it accelerates to its desire speed. But the speed to which the vehicle needs to accelerate cannot be determined with

the information only from one intersection, so we still use the full acceleration and deceleration process to calculate the cost, which is an overestimation of the actual cost. In a word, when links are so short that there is not enough space for a vehicle to accelerate to desire speed and decelerate to zero speed, the approximation method introduced here over estimate the actual costs.

4.4 The constrained eco-routing problem

The vehicle routing problem and eco-routing problem we formulated above only consider one cost of interest at a time. But it is obvious that multiple costs of interest present in an eco-routing problem. And it is possible that people want to consider travel time and environmental costs at the same time. This bring us to the idea of a routing problem with multiple costs of interest.

When there are multiple concerns in one problem, people usually formulate the problem as a multiple objective problem. One common approach to multi-objective problem is to find an optimal solution to a problem with an objective of weighted average of different costs. Applying this approach to the model we developed in previous sections is straightforward, once the weights are known. As stated in the introduction part, another way of thinking the problem is to specify a primary goal of the problem, and then consider other costs as side constraints. To solve this type of problem, we will introduce the linear programming formulation of MDP in the following.

We will still use the MDP developed earlier to describe the problem. Then, we convert the problem to a linear programming. This allow us to add constraints related to costs of interest. Finally, we find the solution to the constrained problem using standard solution method for linear programming.

We first introduce the linear programming formulation of MDP for unconstrained problem. This formulation is similar to “extended TMD-model” described by Kallenberg (1983) in chapter 3 of his book. The term “TMD-model” is used to describe a Markov decision model who uses total reward criterion. The adjective “extended” specifies a model where there is an extra absorbing state. In our case, we should have a set of absorbing states, which represent the destination in our problem.

For each path search problem, there is at least a destination, which is denoted by $D \subset S$. Take our basic model as an example, each state in the model is described by (u, v, t) , where v denotes the current intersection. Then, we should change our model in the following way,

$$\tilde{p}(s'|s, a) = \begin{cases} p(s'|s, a), & \text{if } s = (u, v, t), v \notin D, \\ 0, & \text{otherwise.} \end{cases} \quad (4.5)$$

$$\tilde{c}(s, a) = \begin{cases} c(s, a), & \text{if } s = (u, v, t), v \notin D. \\ 0, & \text{otherwise.} \end{cases} \quad (4.6)$$

Note here $c(s, a)$ is one component of vector $\mathbf{c}(s, a)$ that corresponds to the primary cost of interest. And we still use the same state space S and action space A as we do for the unconstrained problem.

To solve the problem, we use “ALGORITHM VI” introduced by Kallenberg (1983). We rewrite the algorithm using our notations, and call it **Algorithm 2**.

- step 1: Take any vector β such that $\beta_{s'} > 0, s' \in S$.
- step 2: Calculate the optimal solution x^* of the following linear programming problem

$$\begin{aligned}
& \textbf{Minimize} \quad \sum_s \sum_a \tilde{c}(s, a) x_{s,a} \\
& \text{subject to} \quad \sum_s \sum_a (\delta_{s,s'} - \tilde{p}(s'|s, a)) x_{s,a} = \beta_{s'} \quad , s' \in S \\
& \quad \quad \quad x_{s,a} \geq 0 \quad , a \in A_s, s \in S
\end{aligned} \tag{4.7}$$

, where

$$\delta_{s,s'} = \begin{cases} 1 & \text{if } s = s' \\ 0 & \text{otherwise} \end{cases}$$

- step 3: the probability of chosen action a in state s is calculated as

$$d(s, a) = \frac{x_{s,a}}{\sum_a x_{s,a}} \tag{4.8}$$

It has been shown that any feasible solution to the problem has $x_{s,a} > 0$ for exactly one $a \in A_s$ for every $s \in S$. So what we get is a pure and stationary policy (Kallenberg, 1983).

In the linear programming formulation, the decision variables $x_{s,a}$ can be interpreted as the expected number of times action a is chosen in state s . And the vector β is the initial distribution of the system.

This algorithm gives the optimal actions for all the states when the destination is specified. But it does not give expected cost starting from a given state. The value of the objective function is the weighted expected cost from all states to the destination whose weights are give by vector β . When the optimal policy is obtained, it is easy to recover the expected total cost for a given start state from the destination states by assigning zero value at the destination states and backward propagating to the start state.

Based on the linear programming formulation, we can have a constraint method for the problem with multiple costs of interest, by converting objectives concerning specific pollutants into constraints. This is more appropriate in some cases. For example, a driver may want to minimize the fuel consumption while keeping the travel time under a given limit. Linear programming is particularly suitable for this case.

When we impose constraints on expected total cost on one or more costs of interest, a policy that is optimal for all initial states does not exist in general (Kallenberg, 1983). For our problem, we are interested in the constrained optimal policy with respect to a given initial state.

As we have constraints on the expected total cost, we can write the constraints in the following way.

$$\sum_s \sum_a c_{s,a}^q x_{s,a} \leq \tilde{b}_q \quad (4.9)$$

, where $c_{s,a}^q$ is the cost coefficient of q th cost when action a is taken in state s and \tilde{b}_q is the given threshold of q th cost.

Because we only consider a given initial state for the constrained problem, we set the corresponding element of β to be 1, and all the other elements of β to be 0. That is

$$\beta_{s'} = \begin{cases} 1 & \text{if } s' = s_0 \\ 0 & \text{otherwise} \end{cases} \quad (4.10)$$

, where \tilde{s} is the given initial state.

As we add additional constraints to the problem and allow components of β to be zero, it is no longer true that $x_{s,a} > 0$ for exactly one $a \in A_s$ for every $s \in S$.

Consequently, the solution to the problem may not be pure.

To get the optimal stationary policy for a constrained MDP with total expected cost, we use the **Algorithm 3**, which is described below.

- step 1: initialize vector β according to Equation 4.10.
- step 2: Calculate the optimal solution x^* of the following linear programming problem

$$\begin{aligned}
 &\textbf{Minimize} \quad \sum_s \sum_a \tilde{c}(s, a) x_{s,a} & (4.11) \\
 &\text{subject to} \quad \sum_s \sum_a (\delta_{s,s'} - \tilde{p}(s'|s, a)) x_{s,a} = \beta_{s'} & , s' \in S \\
 &\quad \quad \quad \sum_s \sum_a c_{s,a}^q x_{s,a} \leq \tilde{b}_q & , q = 1, 2, \dots, b \\
 &\quad \quad \quad x_{s,a} \geq 0 & , a \in A_s, s \in S
 \end{aligned}$$

, where b is the number of additional cost of interest.

- step 3: the probability of chosen action a in state s is calculated as

$$d(s, a) = \frac{x_{s,a}}{\sum_a x_{s,a}} \quad (4.12)$$

Following the solution obtained by Algorithm 3 will minimize the cost in the objective function on average if a vehicle starts from the given initial state. At the same time, other costs considered as side constraints will be less than or equal to the given threshold on average.

The linear programs in **Algorithm 2** and **Algorithm 3** can be solved using standard solution techniques for linear programs, e.g simplex method or interior points method. It has been shown that interior points method has worst case

polynomial time complexity, while the worst case complexity of simplex method is exponential (see a overview of complexity on linear programming by Megiddo (1987)). In practice, however, both methods, especially the simplex problem, have better performance. For our problem, the matrix is sparse as there are usually 2 or 3 non-zero transition probabilities for each state. This makes it relatively easy to solve the linear program.

Chapter 5

Numerical Examples

5.1 A hypothetical signalized traffic network

We first present three examples in a hypothetical network. Problems with different settings, either in terms of destination or the cost of interest, will be solved in the following. But we only focus on unconstrained problem in this section. The purpose of these examples is to help the readers to better understand the data requirement and the outcomes of the proposed method.

5.1.1 Network layout

In this subsection, problems in a hypothetical signalized traffic network are solved. The network consists of only 12 intersections and is shown in Figure 5.1. These 12 intersections are controlled by traffic signals and indexed from 1 to 12. The traffic signals at intersections 1, 2, 3, 4, 5, 6, 12 are coordinated in order to benefit to traffic propagation along the direction $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 12$. All the other intersections are not coordinated. The other settings, such as the cycle

time and link lengths, will be problem specific.

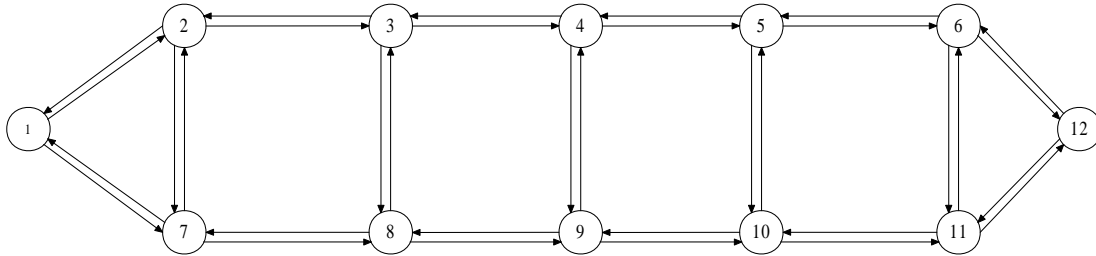


Figure 5.1: The layout of the hypothetical signalized traffic network

5.1.2 Expected total travel time as objective

We first present two examples with the objective to minimize the expected total travel time to the destination. In these examples, we simulate the situation where we get aggregated data from the field. All the intersections are assumed to have 4 possible arrival times and cycle length 4 unit time. The offset between coordinated intersections is 1 unit time. All the links are assumed to have the same travel time characteristics and the travel time distribution is given in Table 5.1.

Furthermore, the same types of intersection movements (e.g. right turn, left turn, and through movement) are assumed to have the same delay distributions, which are given in Table 5.2. For example, the delay of the first cycle time of right turning is given as 0,1 in table 5.2. This means if a vehicle arrives at the intersection at cycle time 0 and chooses to turn right, the possible delays will be 0 and 1 unit time. The corresponding probability is 0.9 and 0.1 respectively, which are given in the right half of the table. U-turn movements are prohibited in this network.

Node 12 and node 6 are set as the destinations in the first and second test

Table 5.1: Link travel time distribution

delay (unit time)	1	2	3
probability	0.1	0.8	0.1

Table 5.2: Intersection delay distributions

	delay				probability			
cycle time	0	1	2	3	0	1	2	3
right	0,1	0,1	0,1	0,1	0.9,0.1	0.8,0.2	0.4,0.6	0.2,0.8
left	2,1	2,1	2,1	2,1	0.2,0.8	0.2,0.8	0.2,0.8	0.2,0.8
straight	0,1,2	0,1	0,1	0	0.1,0.8,0.1	0.3,0.7	0.8,0.2	1

respectively. Origins could be any nodes except the destination nodes. The optimal policies are drawn in Figure 5.2, using arrows in the squares to show the suggested moving directions. The upper and lower parts of the figure give the optimal policies for the first test and second test, respectively. The arrows drawn in one row of squares are the optimal policies at different cycle times when a vehicle arriving at the particular intersection from one approach. For example, the group of arrows circled by the dot oval in test case 1 means that for vehicles from node 3 arriving at node 2 at cycle time 0,1,2,3 should turn left, traveling toward node 7 in order to reach node 12 at minimal expected time cost. For the group of arrows circled by the dash oval, it says vehicles traveling from node 1 to node 2 are advised to go straight to node 3.

Some interesting results can be observed. In case 1, the policies are symmetric as all the link travel time and intersection delays are the same and the network is symmetric. But the optimal costs traveling from node 1 to node 12 are different if different routes are chosen. They are 14.76, 13.99, 13.25, 12.93 unit time for start time 0,1,2,3 at node 1 (delays at node 1 are included) respectively if route $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 12$ is chosen and 15.38, 15.08, 14.58, 14.38 if route

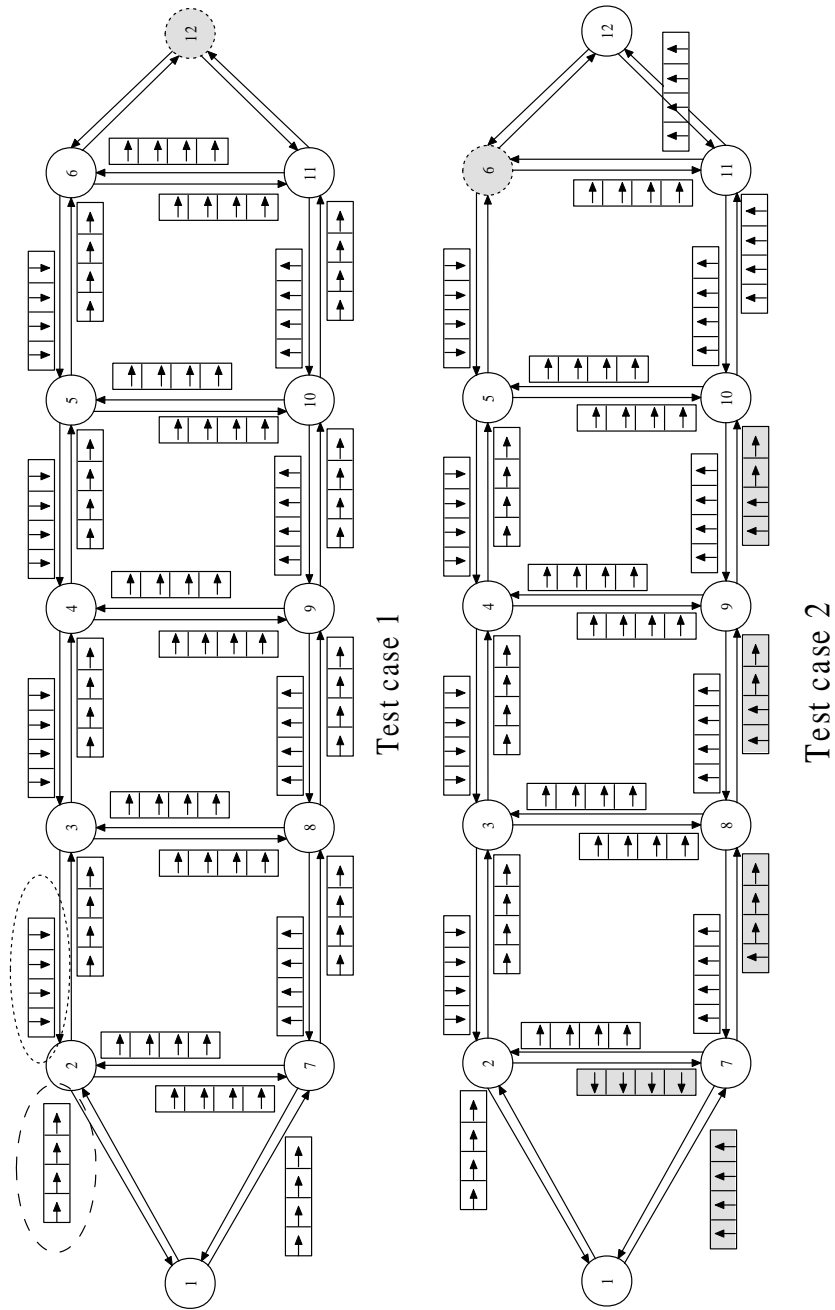


Figure 5.2: Test results in the hypothetical network

$1 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 10 \rightarrow 11 \rightarrow 12$ is chosen. The difference between coordinated and non-coordinated signal controls suggested the coordinated signals can reduce the travel time cost and it is captured by the proposed model.

In test case 2, the optimal policy is not symmetric anymore as the destination changes to node 6. The policies with gray background clearly show the corridor with signal coordination benefits is preferred over the non-coordinated corridor, because travelers are directed to the coordinated corridor. The optimal time costs for traveling from node 1 to node 6 via route $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6$ are 12.54, 11.78, 11.04, 10.72 unit time for arriving time 0, 1, 2, 3 at node 1 respectively and 5 links are traversed. On the other hand, the optimal time costs for traveling from node 7 to node 6 are 12.89, 12.89, 12.48, 12.28 respectively for start time 0,1,2,3 at node 7, where also 5 links are traversed (the actual path taken might be different depending on specific realization of the stochastic process, since the chosen downstream link at an intersection cannot be determined without knowing the arrival time at that intersection, which is the outcome of a random process starting from the origin). Although both paths include 5 links, the expected travel times of the first path (from node 1 to node 6) are smaller than those of the second path (from node 7 to node 6). The time saving comes from the time costs of different turning movements as well as the benefits of coordinated signal control. Furthermore, the second test case shows that the route choice can be different depending on the arriving time at the intersections. For example, a vehicle traveling from node 8 to node 9 should choose to turn left if it arrives at node 9 at the first two cycle times and choose to go straight if arriving at node 9 during the last two cycle times.

5.1.3 Expected fuel consumption as objective

In this example, we still use the hypothetical network shown in Figure 5.1. The objective is to find the optimal paths to the destination with least expected fuel consumption. Different from last example, U-turn is allowed in this example. We will also demonstrate how time aggregation can be used to decrease the size of state space.

With the geometry shown in Figure 5.1. All the links are assumed to have the same length of 800 *feet*. All the U-turns at intersections are assumed to have the same delay characteristics as corresponding left turns. Intersection movements are assigned to phases according to Figure 5.3. Figure 5.3(a) corresponds to *intersection 1* and *12*. Figure 5.3(b) gives the phase assignment for *intersection 2* to *6*. And Figure 5.3(c) is for *intersection 7* to *11*. At coordinated intersections, the reference phase is *phase 2*.

The cycle lengths for all the intersections are assumed to be 60 seconds. But to reduce the size of state space and thus the computational cost, it is assumed there are 4 possible arriving time periods at the intersections. From the start of green time of *phase 2*, the first 10 *seconds* is defined as the starting period of green (GS); the next 20 *seconds* defined as the ending period of green (GE); the period from 31 to 55 *seconds* is the starting period of red (RS); and the last 5 *seconds* is ending period of red (RE). The time aggregation and signal phases are shown in Figure 5.4.

The logic for using this four periods is that transition probability and delays may be the similar within the same time period but quite different across these time periods. For example, if a vehicle arrives during GS, there might be queue in front, forcing the vehicle stops at the intersection. But if a vehicle arrives during GE, there is usually no queue at the intersection, and the vehicle can pass the

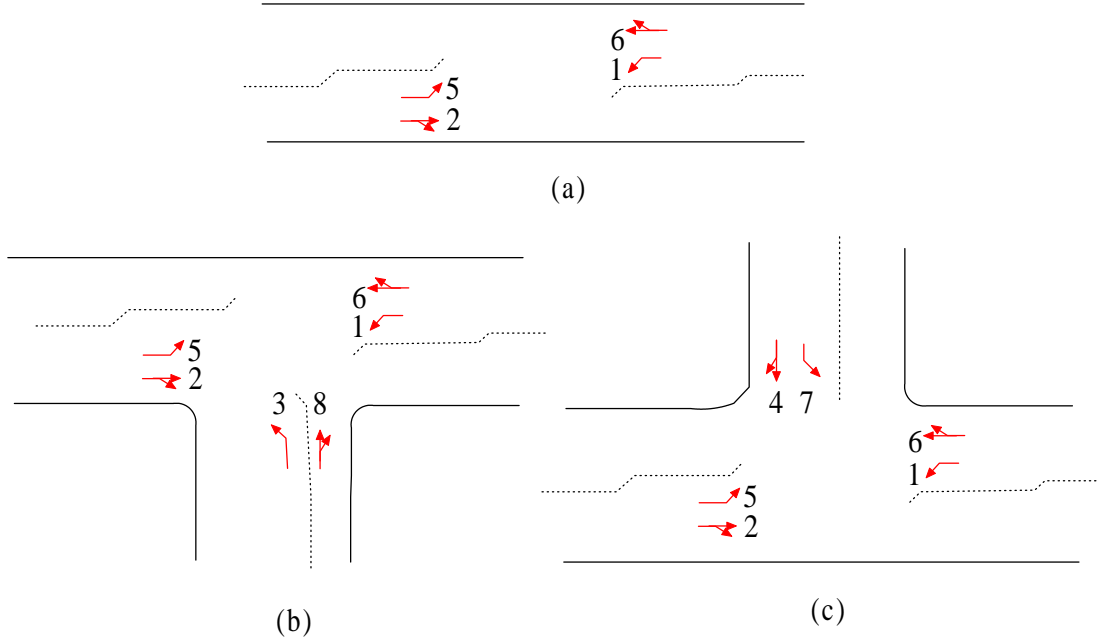


Figure 5.3: Intersection phase settings

intersection without a stop.

Given these four time intervals of a cycle (temporal granularity), aggregated transition probability and delay distribution can be estimated from raw high-resolution traffic data, as shown in Section 3.3.2. In this example, these aggregated data are assumed to follow the distributions given in Table 5.3, Table 5.4, Table 5.5. and Table 5.6.

Table 5.3 gives the transition probabilities between different time periods at coordinated intersections. For non-coordinated intersections, the transition probabilities are calculated based on the assumption that the possibility to arrive at downstream intersection at any second is uniformly distributed (Table 5.4).

Table 5.5 and Table 5.6 are the expected intersection delays and probabilities to stop at intersections for different arriving time periods at intersections. When

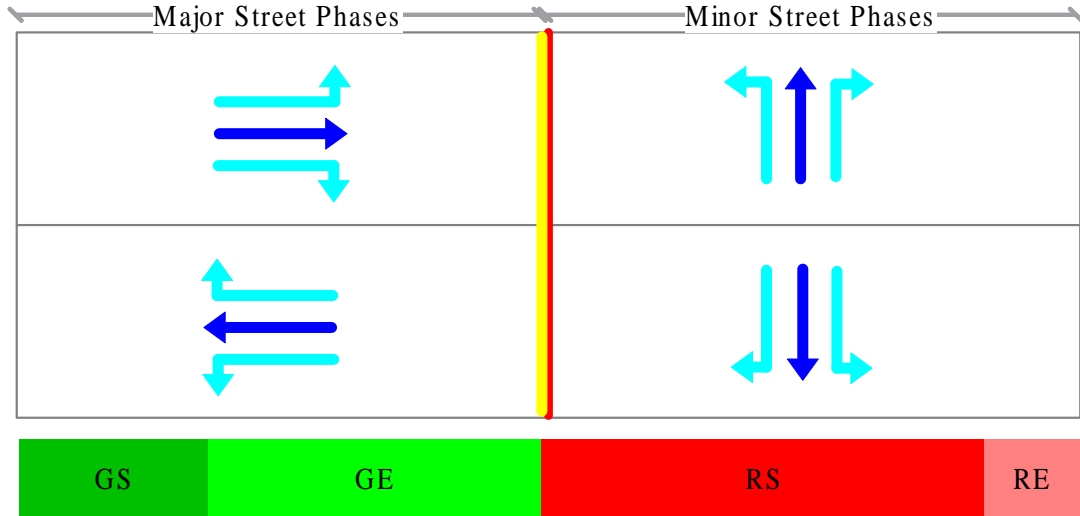


Figure 5.4: Intersection signal phases and time periods

properly configured, intersection delays for coordinated phases are usually smaller than those of other phases, and the probabilities to stop at intersections along the coordinated directions are usually lower. But in this example, it is assumed that these probabilities are the same for all the intersections. This assumption allows one to more clearly see the effects of different transition probabilities due to signal coordination, which is equivalent to see the consequences of better traffic propagation along coordinated directions.

In this example, we still set *intersection 12* as the destination. The cost of interest is fuel consumption. Vehicle characteristics and the corresponding link divisions are given in Table 5.7. With these information, vehicle trajectories are generated according to our assumptions given in Section 4.3. These trajectories are used as inputs to the microscopic emission model CMEM (Barth et al., 2000). Fuel consumption parameters are calculated with default model parameters and

Table 5.3: Transition probability at coordinated intersections

state s	state s'	GS	GE	RS	RE
GS		0.6	0.3	0.1	0
GE		0.2	0.6	0.1	0.1
RS		0.6	0.2	0.1	0.1
RE		0.6	0.3	0.1	0

Table 5.4: Transition probability at non-coordinated intersections

state s	state s'	GS	GE	RS	RE
GS		0.167	0.333	0.417	0.083
GE		0.167	0.333	0.417	0.083
RS		0.167	0.333	0.417	0.083
RE		0.167	0.333	0.417	0.083

given in Table 5.8.

Although only fuel consumption is used in this example, different pollutants (e.g. HC, NO_x, CO) can also be used as the cost of interest in the same manner. Please note the value of fuel consumption parameters are the same for all the links in this example as we assume the same link length and desire speed on all the links. In a real traffic networks, these cost parameters may be different across the links as link lengths and desire speed are different, which can be seen later.

The most fuel efficient policy can be found using value iteration method (Algorithm 1). And the results are shown in Figure 5.5. In this figure, the optimal policy at each intersection indicated by arrows depending the arrival times, i.e. GS, GE, RS, and RE of *phase 2* at each intersection. For example, if one arrives

Table 5.5: Expected intersection delays (seconds)

Time	Phase	1	2	3	4	5	6	7	8
GS		7	2	18	7	8	4	18	7
GE		3	0	8	3	3	1	8	3
RS		25	18	4	3	26	18	4	3
RE		9	6	1	1	11	8	1	1

Table 5.6: Stop probability at intersections

Time	Phase	1	2	3	4	5	6	7	8
GS		1	0.2	1	1	1	0.3	1	1
GE		1	0.1	0.9	1	1	0.1	0.9	1
RS		1	0.8	0.2	1	1	1	0.2	1
RE		1	0.8	0.2	1	1	0.8	0.2	1

at *intersection 3* from *intersection 2*, the optimal policy is to go to *intersection 4*, indicated by 4 arrows pointing to the right.

As the network is symmetric, most of the policies are stationary at intersections. But if one arrives at *intersection 7* from *intersection 8*, making right turn during the first 2 time periods and making U-turn during the last 2 time periods are suggested by the algorithm. Furthermore, if one arrives at *intersection 1* from *intersection 2*, it is advised to make a U-turn, while going forward is better if one arrives at *intersection 1* from *intersection 7*. These two cases show the tendency to use the coordinated route.

The benefit of signal coordination is more clearly seen by looking at the expected costs. Two routes can be followed to travel from *intersection 1* to *intersection 12*, i.e. the **upper route** ($1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 12$) and the

Table 5.7: Vehicle characteristics and link segment division

desire speed (\dot{x})	30 <i>mph</i>
acceleration rate (\ddot{x}_a)	3.2 <i>feet/s²</i>
deceleration rate (\ddot{x}_d)	10 <i>feet/s²</i>
first segment (x_1)	302.5 <i>feet</i>
middle segment (x_2)	400.7 <i>feet</i>
end segment (x_3)	96.8 <i>feet</i>

Table 5.8: Fuel consumption parameters

$c_{12}(g)$	$c_{22}(g)$	$c_{32}(g)$	$c_{33}(g)$	$c_{11}(g)$	idle rate (<i>g/s</i>)
4.65	5.23	1.16	2.41	16.97	0.581

lower route ($1 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 10 \rightarrow 11 \rightarrow 12$). The costs of upper route are 75.55 *g*, 74.80 *g*, 92.97 *g*, and 86.25 *g* for GS, GE, RS, and RE respectively. These numbers are 98.52 *g*, 96.93 *g*, 114.80 *g*, and 109.21 *g* for the lower route. This clearly shows that the route with traffic signal coordination provides high fuel efficiency. This demonstrates the savings of fuel from signal coordination and how this can be captured by the proposed model, which is designed to choose the most eco-friendly routes in a signalized traffic network.

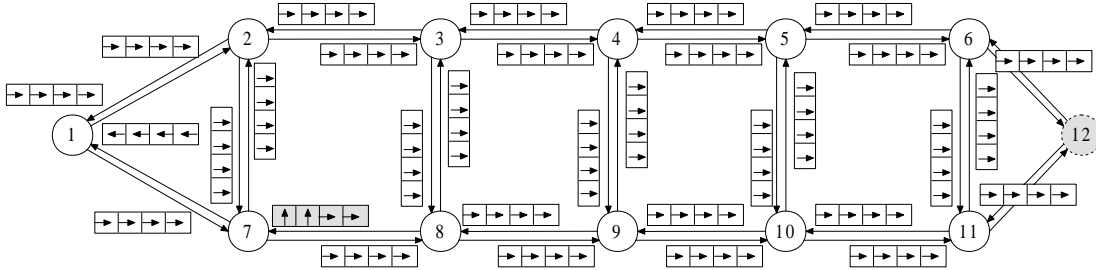


Figure 5.5: Optimal fuel consumption policy

5.2 A Real world signalized traffic network

Next, we will use data from a real world signalized traffic network to demonstrate the methods proposed in the dissertation. The traffic signal data are obtained from the City of Pasadena, California, United States. As the high-resolution vehicle actuation data are not available for this site, queuing delays (δ_q) are not considered. Only delays caused by red lights (δ_r) are considered in the following.

In the following, we first give out the configurations of the experiment. Then, we will do a numerical example using expected travel time as the objective. In this particular example, the result from the proposed method will be compared against that from the traditional shortest path algorithm that uses link travel times as costs. We use the most traditional shortest path algorithm for two reasons: 1) there is no known shortest algorithm that can deal with vehicle actuated traffic signals; 2) the traditional shortest path algorithm is well understood by people, so it should be easier for people to understand the results if it is used as the baseline. Following this example, several more examples will be given, in which environmental objectives are used to demonstrate the proposed method in this dissertation.

5.2.1 Test site settings

A network consisting of 20 intersections in downtown Pasadena is chosen. 19 of these 20 intersections are controlled by traffic signals except for the intersection of Cordova St and Oak Knoll Ave, which is controlled by stop signs for the north-south approach. In Figure 5.6, the network used in the experiment is colored by blue lines. All the signalized intersections are indicated by green balloons, and the intersection controlled by stops is shown by red balloon.

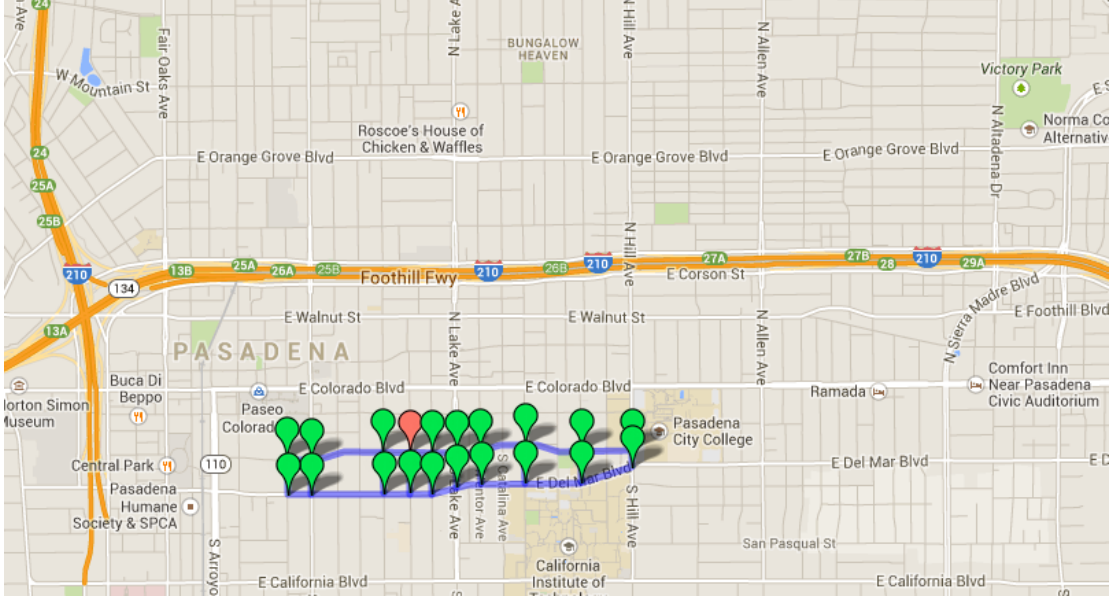


Figure 5.6: The location of test site in the City of Pasadena, CA (Google Maps)

The configurations of the traffic signals are given in Figure 5.7. Each circle represents an intersection and the intersections are numbered from 1 to 20. Intersection 4 is the non-signalized intersection mentioned earlier. The speed limits for all the east-west direction links are 35 *mph* and all the north-south direction links are 25 *mph*. Please also note that Hudson Ave and Mentor Ave are one way roads.

We use second-by-second state space model in this example. There are 20 intersections in this network. Each intersection has 2 to 3 arriving directions. For each arriving direction, there are 60 or 80 possible arrival times, depending on the cycle length at the intersections. So the size of the state space should be less than $20 \times 3 \times 80 = 4800$.

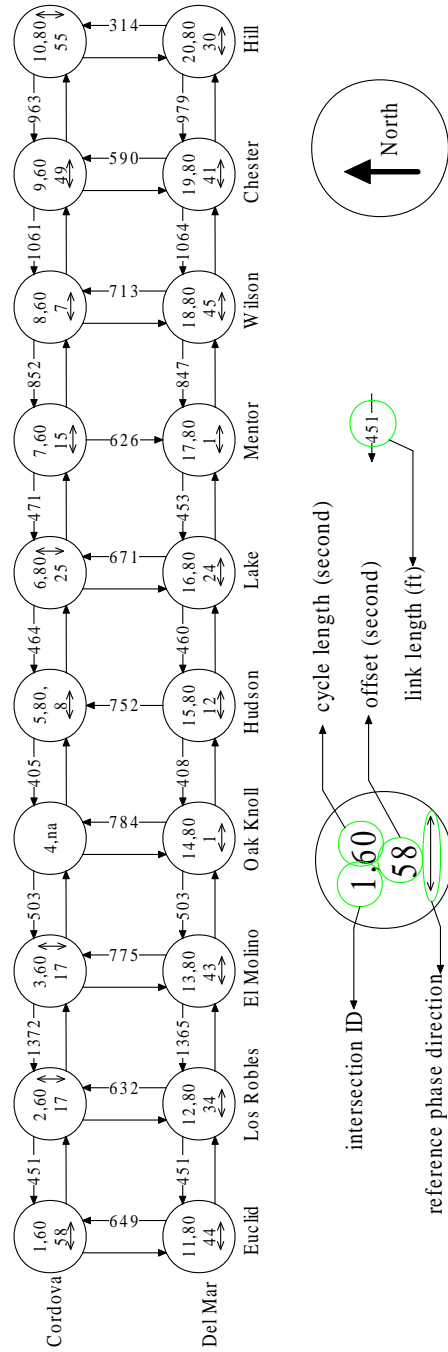


Figure 5.7: Network configurations

The coordination directions of signalized intersections are given by arrows in the circles. It can be seen that all the intersections along Del Mar Blvd are coordinated while it is not the case for the intersections on Cordova St. Signal event data from four working days are used to generate red light delay distributions. These four days are Nov. 21, Nov. 22, Nov. 23 and Nov. 26, 2012. Data during morning peak hours from 7:00 to 9:00 on these days are used. The signal plans are the same during these times. Some sample data are given in Table 5.9 and the meanings of events are given in Table 5.10. Each row in Table 5.9 represents an event of traffic signal status change. For example, the first row in Table 5.9 means a pedestrian light for phase 2 of intersection 1 changes to “walk” at 00:00:18.603 on November 21th, 2012.

Table 5.9: Sample signal data

Time stamp	Intersection	Event	Phase
2012-11-21 00:00:18.603	1	3051	2
2012-11-21 00:00:18.603	1	3031	2
2012-11-21 00:00:18.603	1	3033	4
2012-11-21 00:00:18.603	1	3051	6
2012-11-21 00:00:18.603	1	3031	6

Given network configuration shown in Figure 5.7 and signal information like in Table 5.9 and 5.10, a red light delay distribution at each possible arrival time (second by second) for each intersection movement can be estimated. An example of such distribution is shown in Figure 5.8. It is for the movement from intersection 16 to intersection 18 via intersection 17, which is the coordinated movement at intersection 17. There are two peaks in Figure 5.8. The peak around 16 seconds corresponds to the cycles where there is no pedestrian calls during the cycle, while the other peak has a higher value of delays due to the pedestrian calls. Given red

Table 5.10: Signal events

Event	Explanation
3031	Phase Vehicle Display Changed To Green
3032	Phase Vehicle Display Changed To Yellow
3033	Phase Vehicle Display Changed To Red
3051	Ped Changed To Walk
3052	Ped Changed To Flash Don't Walk
3053	Ped Changed To Don't Walk

light distributions, the transition probabilities between different states can be calculated by Equation 3.10, ignoring queuing delays.

5.2.2 Expected total travel time as objective

In the first example, the objective is to minimize the expected total travel time. And we set *intersection 20* (Del Mar Blvd at Hill Ave) as the destination. By solving the problem using the value iteration method described earlier, the optimal actions to get the destination with minimal expected total travel time are obtained for all the states.

To see the advantage of the proposed method, we will compare the proposed method against the traditional shortest path algorithm. We solve the shortest path problem in terms of travel time using the traditional shortest path algorithm, more specifically, the Dijkstra's algorithm. As mentioned at the beginning of the chapter, the traditional shortest path algorithm is used mostly because there is no known method for calculating optimal path when vehicle actuated traffic signals are present. We ignore the traffic signal when calculating the optimal path when using traditional shortest path algorithm.

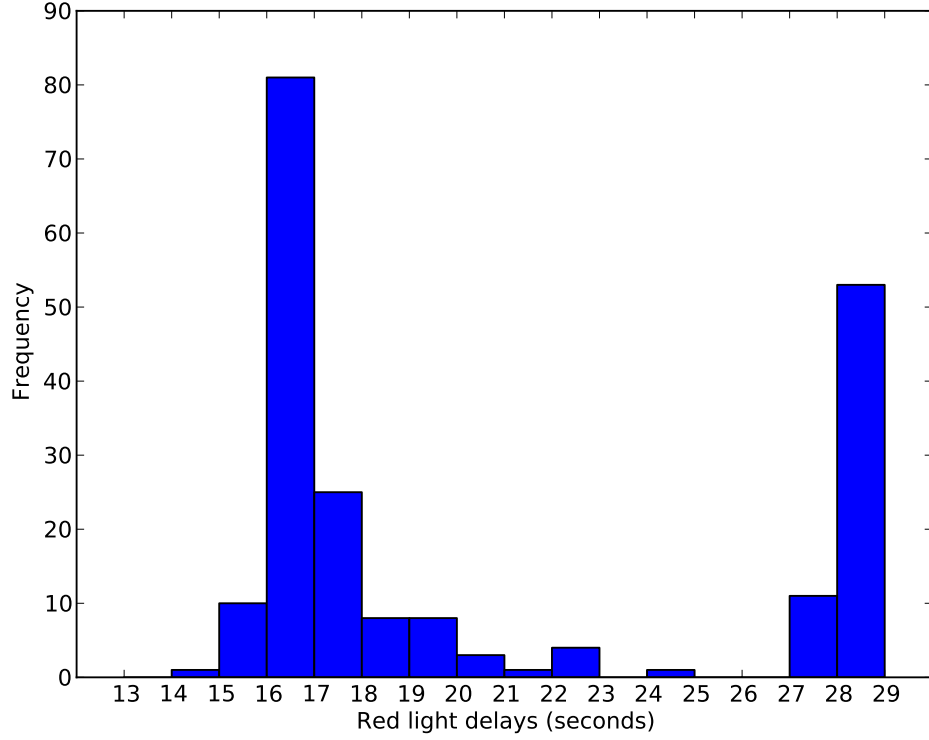


Figure 5.8: An example of red light delay distribution

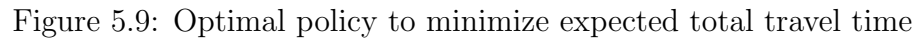
In this example, the optimal policy will be calculated using the method introduced in Chapter 3, by assuming static link travel time and intersection delays only including red light delays. When applying the Dijkstra's algorithm, we use the free flow link travel time on links as link costs. Intersection delays are not considered when calculating the optimal path by Dijkstra's algorithm. But they will be considered when calculating the actual travel time using the virtual probe approach described below.

Because of the randomness from traffic condition, it is necessary to run the

experiment for many times to compare two methods. As we have collected large amount of historical data, a virtual probe approach is going to be employed. In this approach, an imaginary vehicle travels in the network in the past, and corresponding intersection delays are calculated using available signal information. This approach works in the following way.

To use the optimal policy from the proposed method, let a vehicle starts to travel from a given intersection at a given time. The time for the vehicle to arrive at the next intersection can be calculated by assuming the vehicle traveling at free flow speed. Then, the optimal action is chosen based on the arrival time at the next intersection. Given the signal data is known at that time, the corresponding intersection delays can be calculated, and consequently, the following start time for the next link is known. This process can be repeated until the vehicle reaches to the destination. To follow the optimal path from Dijkstra's algorithm, the process is similar except there is no action choice at each intersection based on the arrival time. The vehicle just follows the optimal path given by the Dijkstra's algorithm.

The travel time for a specific run according to the optimal policy from the proposed method or the optimal path from the Dijkstra's algorithm can be calculated by taking the difference between trip start time and trip end time. By repeating this process for many times using historical data, the average travel time by following the optimal policy from the proposed method can be compared against the average travel time by following the optimal path given from Dijkstra's algorithm. It should be also pointed out that the static travel time on links and intersection delays only including red light delays are also the assumptions that are used when we calculate the optimal policy.



According to the optimal path or optimal policy, we carry out 6 experiments

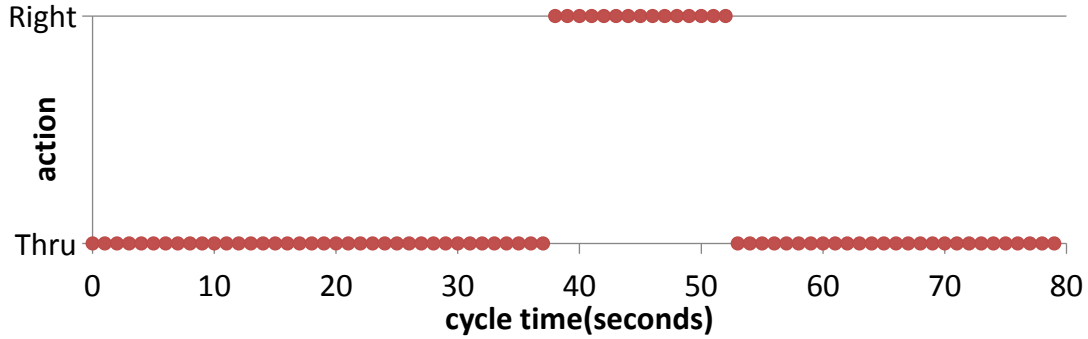


Figure 5.10: Time dependent actions at intersection 6

using virtual probe approach introduced above. Data during morning peak hours of Nov/21/2012, Nov/22/2012, and Nov/23/2012 are used. For each experiment, we imagine that vehicles start to travel from *intersection 1* at every minute during a 45 minute time period. So there are 45 runs for each experiment. For each day, we have conducted two such experiments. One starts from 7:11 am and the other starts from 8:11 am. The average travel time reductions from tradition shortest path algorithm by using our proposed method are shown in Table 5.11.

Table 5.11: Average travel time reduction by using proposed method

Time	Nov/21/2012	Nov/22/2012	Nov/23/2012
7:11 am	-5.82%	-18.38%	-2.70%
8:11 am	-2.83%	-5.82%	-2.74%

It can be easily seen that the proposed method is always better than the traditional shortest path algorithm on average. The savings in average travel time are mostly between 2% to 5%. There is an exception on November 22, 2012 between 7:11 am to 7:56 am, when the saving is more significant. After examining

the data, it turns out the reason for this is that the green time is unusual long for north-south direction at *intersection 6*, which last more than 2 minutes, for some cycles during that time. This is the direct cause of long delays if a vehicle follows the optimal path given by Dijkstra’s algorithm. The cause of unusual long green time is unknown at this moment. But non-coordinated phases at intersections, as in this case, are prone to this kind of extra delays.

The improvement mainly comes from the additional information from traffic signals. The traditional shortest path algorithm ignores such information when searching for the optimal path, while the proposed method explicitly accounted for this factor. It should also be pointed out that the proposed method is only better on average, which means the actual travel time may be longer by following the optimal policy for some runs.

5.2.3 CO emission as objective

In the previous hypothetic traffic network, we have calculated the optimal policy using fuel consumption as objective. In the Pasadena network, we do the same calculation and find out that the optimal policy using fuel consumption as objective is almost the same as the optimal policy when using travel time as objective.

This is easy to understand as less travel time in a signalized traffic network usually results from less stops at intersections, which also results in less fuel consumption and pollutant emissions. In addition, the free flow speed in urban arterial networks is around 35 *mph*, which is quite fuel efficient. So the optimal policy with regards to travel time is also good for minimizing the fuel consumption. But this is not always true for all the pollutants. Next, we will show the optimal policy for minimizing CO emission is somewhat different from that for minimizing travel time in the same network.

In this example, we still assume the free flow speed on links is the same as post speed limit on the links, which are 25 *mph* for north-south directions and 35 *mph* for east-west directions. We also assume the acceleration rate is 4 *ft/s*² and deceleration rate is 10 *ft/s*². Based on these assumptions, the CO emission parameters are calculated using CMEM model and given in Table 5.12.

Table 5.12: CO emission parameters

Speed	$c_{11}(g)$	$c_{12}(g)$	$c_{32}(g)$	$c_{33}(g)$	cruise rate (<i>g/s</i>)	idle rate (<i>g/s</i>)
35 <i>mph</i>	1.0326	0.0950	0.0380	0.0394	0.0148	0.0041
25 <i>mph</i>	0.5254	0.0300	0.0120	0.0229	0.0065	0.0041

As usual, we set *intersection 20* (Del Mar Blvd at Hill Ave) as the destination. And the optimal policy for CO emission to start from the *intersection 1* is given in Figure 5.11. Red arrows again show the optimal policy when going east from *intersection 1* and green arrows give the optimal policy when going south.

Comparing Figure 5.11 to Figure 5.9, it can be seen that the optimal policies for CO emission and travel time are similar to each other. But if we take a closer look at the optimal action at *intersection 2*, it can be seen that the coordinated route is used more when CO emission is used as objective (Figure 5.12).

The difference in optimal policy for travel time and co emission may be caused by the different ratios of costs associated with stop and free flow traveling. What can be observed from Table 5.12 is that the emission rate of CO at cruise mode is quite low. The CO emission per second at cruise mode is only about 1.5% of the acceleration process (c_{11}). As comparison, this number is more than 3% for the fuel consumption, which means CO emission is more sensitive to stops.

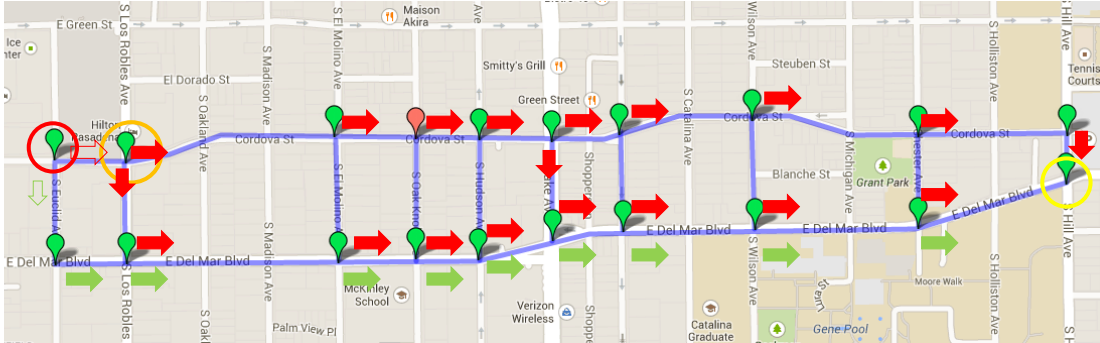


Figure 5.11: Optimal policy to minimize expected total CO emission

5.2.4 The constrained problem

Previous examples have shown the possibility to find the optimal paths with regards to various objectives in a real world traffic network when traffic signal information is incorporated. But all these examples only have a single objective at a time. If it is necessary to consider more than one cost of interest at the same time, we need apply the method introduced in Section 4.4.

One prerequisite of solving the constrained problem is to determine the constraint constants (\tilde{b}_q). In this example, we first calculate the optimal value for each cost of interest and then add some buffer to each one of the cost that is used as constraint to get the constraint constants.

We still use *intersection 20* as destination. The optimal costs for starting from *intersection 1*, arriving at *intersection 2* at cycle time 1, to get to the destination are given in row 1 of Table 5.13. These optimal values are obtained by solving unconstrained problems for each cost of interest.

Next, we solve a problem with minimizing travel time as the primary objective.

*the optimal expected travel time given the constraints on other costs.

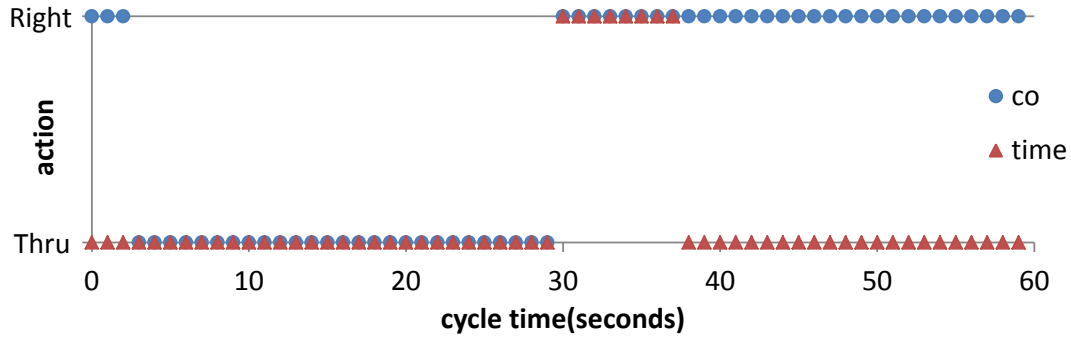


Figure 5.12: Time dependent actions at intersection 2

Table 5.13: Optimal values and constraints constant of different costs

	time (s)	fuel (g)	CO (g)	CO ₂ (g)	HC (g)	NO _x (g)
optimum	211.8404	234.0996	6.5032	731.203	0.337	0.5695
constraint	224.9417*	260	6.7	735	0.35	0.59

The costs of other pollutants are used as constraints. The constraint constants are given in row 2 of Table 5.13. It can be seen that we are no longer able to achieve the optimal travel time when constrained by other costs. But it is also obvious that the difference between the constrained optimal value and the true optimal value is quite small. This means the costs considered here do not contradict to each other a lot in our problem settings.

Another interesting thing to mention is that the optimal solution for the constrained problem is no longer pure. We get randomized action at *intersection 2*, as shown in Figure 5.13. It is best to go straight about 90% of the time and turn right about 10% of the time when arriving at that intersection at cycle time 1.

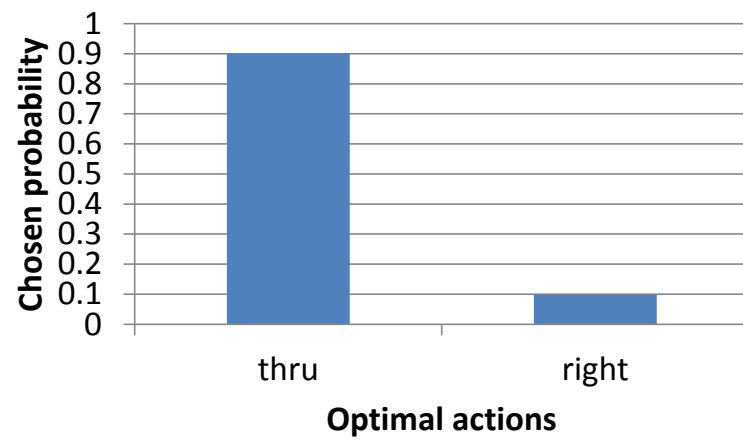


Figure 5.13: Randomized action at intersection 2 at the first cycle time

Chapter 6

Summary and Future Research

6.1 Summary

In this dissertation, we have studied various path search problems for vehicle routing explicitly considering the traffic signal information. This is motivated by the increasing availability of high-resolution traffic data from signalized traffic network. The information from high-resolution traffic data includes detailed signal and vehicle actuation events, i.e. every signal status change and every vehicle actuation. Although it has been shown that these information is very useful in traffic status estimation and traffic operation optimization, little is known how they can help with path search problems.

Path search problems, or widely known as shortest path problems, are of great importance in transportation science, but there were very limited efforts on the path search problems considering traffic signals, especially when actuated traffic signals are used. This research is aimed to fill this gap.

The advantage of this research is the availability of the high-resolution traffic data. The difficulty comes from the randomness in costs brought in by actuated

traffic signals. In addition, we should also consider the correlations between costs at adjacent intersections because of traffic propagation and signal coordination. The signal coordination in urban arterial networks is designed to promote better traffic propagation and thus reduce travel time. But it is challenging to quantitatively describe this benefit and incorporate it into a path search algorithm.

In this dissertation, the path search problem is formulated as a Markov decision process (MDP) with expected total cost as optimality criterion. By exploiting the cyclic property of traffic signals, an infinite horizon and finite MDP is employed. The states generating from destination is set to be absorbing set. The state space of the model depends on the geometry of the traffic network, as well as the time discretization scheme. Based the specific application requirements, different time discretization schemes can be used to accommodate the computational resource constraints.

Transition probabilities between states of coordinated intersections are constructed using signal control parameters, including cycle length and offset, as well as delay information, which can be estimated from high-resolution traffic signal data. We have also extended our formulation so that real-time traffic signal information can be used when available. The method to estimate delays at intersections are given in Section 3.5 of Chapter 3. Sample delay distributions are shown using field data collected from TH-55 in Minnesota.

We have first studied the problem where travel time is the cost of interest. Stage cost consists of red light delay, queuing delay and link travel time. This is the same information with which we constructed the transition probabilities. The objective is to minimize the expected total cost to the destination.

To solve the problem, we have used value iteration method. We choose the value iteration method because it is one of the standard methods that are used to

solve MDP and very flexible in terms of the solution process. Different thresholds can be used to control the accuracy of the solution, as well as the computational cost.

Next, we have studied the problem where the costs of interest are environmentally related. This type of problems is called “eco-routing” problem in the literature and has received quite some attention recently. To calculate environmental related costs, we have proposed to use well-developed microscopic vehicle emission models, such as CMEM, VT-Micro. We have also developed a way to integrate the microscopic vehicle emission models into our MDP-based path search algorithm.

As there can be different costs of interest in a path search problem, a natural question to ask is whether it is possible to consider several of them at the same time. This leads us to a multi-objective path search problem. One common approach to multi-objective problem is to find an optimal solution to a problem with an objective of weighted average of different costs. Applying this approach to the method we developed for single objective is straightforward, once the weights are known.

Instead of the weighting method, in this dissertation, we use a constraint based method by converting some of the costs of interest into constraints. This is done by introducing the linear programming (LP) formulation of MDP. Although the method to solve an MDP problem using LP is usually expensive, it has some merits in its own. First, this allows the use of some well developed and ready to use software tools to solve the problem. Second, the LP formulation allows extra constraints to be added to the problem. This allows a constraint based approach to address a multi-objective path search problem, which is the major reason for us to introduce LP formulation of MDP into this study.

Different from traditional short path algorithms, the output of the proposed methods is an optimal policy instead of a single path. An optimal policy prescribes how to choose the best action given the realization of a stochastic process, i.e. traveling in a traffic network with stochastic traffic dynamics in our case.

Finally, we have demonstrated the proposed methods using a hypothetical traffic network and a real world traffic network. We have solved the problems with travel time as objective, as well as the problems with various environmental costs as objective. For the problem where travel time is used as objective, we have compared the result from our algorithm to that from the traditional shortest path algorithm. The comparison demonstrated that our algorithm is always superior on average when vehicle actuated traffic signal presents. We have also solved a constrained problem where more than one costs of interest are considered. Using these examples, we have shown the data requirements proposed model and demonstrated the features of the proposed model. By using the real world example, we have also showed the applicability and readiness of our methods to real world problems.

6.2 Future research

This dissertation serves as an initial step to explore the possibility to make use of detailed traffic signal and states information in a path search problem. There are many possible ways for improvement. In this dissertation, we have used value iteration method to solve an MDP. When implementing the value iteration algorithm, we update all the state value at each iteration. As a matter of fact, it is possible to update a portion of the states that is more relevant given some additional information at each iteration. This results in a class of heuristic algorithms, which are worthy of exploration. We have introduced linear programming approach for

solving constrained MDP. But this results in the possibility of randomized policy in the optimal solution. More exploration is needed to interpret the randomized action and how to make use of them in practice.

Besides these two ways to solve an MDP, there are other possible solution methods, such policy iteration, LAO* (Hansen and Zilberstein, 2001), and RTDP (Barto et al., 1995), etc. Which method is more appropriate for the proposed model needs to be examined. Furthermore, how to collect and prepare the required input data also becomes a non-trivial issue for a large network.

A lot of research on “eco-routing” problems used historical GPS data, which we haven’t used when estimating environmental costs. It should be possible and useful to include the GPS data in our proposed method. To do this, we first need to collect a fair amount of GPS based vehicle trajectories, and then associate these data with signal status and traffic state information. Then, instead of use pre-determined vehicle parameters, such as free flow speed, constant acceleration and deceleration rates, we can use real world vehicle trajectories for cost estimation.

The proposed methods also need further investigation in a large size real world traffic network. Although we have tested our algorithm in a real world traffic network in this dissertation, the size of the network is small, due to the constraint of time and data availability. It will be interesting to see the performance of the proposed methods in a larger size of real world network. Furthermore, as we are studying a stochastic problem, the realization of a stochastic will be different from one run to another. For the purpose of statistical analysis of the algorithm, Monte Carlo simulation may be employed if large scale of real world tests are not possible.

Finally, we have implicitly assumed that routing guidance to individual vehicles have no impacts on traffic conditions. But this is not true when many of the

vehicles on roads follow the same instruction. The distributions of signal durations actuated by vehicle arrivals, as well as the queuing dynamics, may no longer be the same. How to deal with this require a great deal of efforts in the future.

References

- Ahn, K. and Rakha, H. (2008). The effects of route choice decisions on vehicle energy consumption and emissions. *Transportation Research Part D: Transport and Environment*, 13(3):151 – 167.
- Ahn, K., Rakha, H., Trani, A., and Van Aerde, M. (2002). Estimating vehicle fuel consumption and emissions based on instantaneous speed and acceleration levels. *Journal of Transportation Engineering*, 128(2):182–190.
- Ahuja, R., Orlin, J., Pallottino, S., and Scutellà, M. (2002). Minimum time and minimum cost-path problems in street networks with periodic traffic lights. *Transportation Science*, 36(3):326–336.
- Barth, M., An, F., Younglove, T., Scora, G., Levine, C., Ross, M., and Wenzel, T. (2000). Development of a comprehensive modal emissions model. Technical Report NCHRP PROJECT 25 - 11, National Cooperative Highway Research Program.
- Barth, M., Boriboonsomsin, K., and Vu, A. (2007). Environmentally-friendly navigation. In *Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE*, pages 684–689.

- Barto, A. G., Bradtke, S. J., and Singh, S. P. (1995). Learning to act using real-time dynamic programming. *Artificial Intelligence*, 72(12):81 – 138.
- Bertsekas, D. (1987). *Dynamic Programming: Deterministic and Stochastic Models*. Prentice-Hall.
- Bertsekas, D. (1995). *Dynamic programming and optimal control*. Athena Scientific Belmont, MA.
- Bertsekas, D. and Tsitsiklis, J. (1991). An analysis of stochastic shortest path problems. *Mathematics of Operations Research*, pages 580–595.
- Bonet, B. (2007). On the speed of convergence of value iteration on stochastic shortest-path problems. *Mathematics of Operations Research*, 32(2):365–373.
- Boriboonsomsin, K., Barth, M., Zhu, W., and Vu, A. (2012). Eco-routing navigation system based on multisource historical and real-time traffic information. *Intelligent Transportation Systems, IEEE Transactions on*, 13(4):1694–1704.
- Caldwell, T. (1961). On finding minimum routes in a network with turn penalties. *Communications of the ACM*, 4(2):107–108.
- Chen, A., Zhou, Z., and Ryu, S. (2011). Modeling physical and environmental side constraints in traffic equilibrium problem. *International Journal of Sustainable Transportation*, 5(3):172–197.
- Chen, Y. and Yang, H. (2000). Shortest paths in traffic-light networks. *Transportation Research Part B: Methodological*, 34(4):241–253.
- d’Epenoux, F. (1960). Sur un probleme de production et de stockage dans laléatoire. *Revue Francaise Recherche Oprationelle*, 14:3–16.

- d'Epenoux, F. (1963). A probabilistic production and inventory problem. *Management Science*, 10(1):98–108.
- Derman, C. (1970). *Finite State Markovian Decision Processes*. Academic Press, Inc., Orlando, FL, USA.
- Dijkstra, E. (1959). A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271.
- Easa, S. (1985). Shortest route algorithm with movement prohibitions. *Transportation Research Part B: Methodological*, 19(3):197–208.
- Eaton, J. and Zadeh, L. (1962). Optimal pursuit strategies in discrete-state probabilistic systems. *Journal of Basic Engineer*, 84:23–29.
- El-Shawarby, I., Ahn, K., and Rakha, H. (2005). Comparative field evaluation of vehicle cruise speed and acceleration level impacts on hot stabilized emissions. *Transportation Research Part D: Transport and Environment*, 10(1):13–30.
- EPA (2008a). Annual energy outlook 2008 with projections to 2030. Technical Report Report DOE/EIA-0383, United States Environmental Protection Agency.
- EPA (2008b). Inventory of U.S. greenhouse gas emissions and sinks: 1990-2006. Technical report, United States Environmental Protection Agency.
- EPA (2010). Using MOVES in project-level carbon monoxide analyses. Technical Report EPA-420-B-10-041, United States Environmental Protection Agency.
- EPA (2012). *Motor Vehicle Emission Simulator (MOVES) - User Guide for MOVES2010b*. United States Environmental Protection Agency.

- Ericsson, E., Larsson, H., and Brundell-Freij, K. (2006). Optimizing route choice for lowest fuel consumption potential effects of a new driver support tool. *Transportation Research Part C: Emerging Technologies*, 14(6):369 – 383.
- Federal Highway Administration (2008). Traffic signal timing manual. Technical report, Federal Highway Administration.
- Ford, L. (1956). Network flow theory.
- Frey, H. C., Roupail, N. M., Unal, A., and Colyar, J. D. (2001). Emissions reduction through better traffic management: An empirical evaluation based upon on-road measurements. Technical Report FHWA/NC/2002-001, Center for Transportation and the Environment Department of Civil Engineering North Carolina State University.
- Frey, H. C., Zhang, K., and Roupail, N. M. (2010). Vehicle-specific emissions modeling based upon on-road measurements. *Environmental Science & Technology*, 44(9):3594–3600.
- Hammarström, U. (1999). Mätning och simulering av bilavgaser körning med och utan husvagn i laboratorium och på väg. Technical Report 856, VTI meddelande.
- Hansen, E. A. (2011). Suboptimality bounds for stochastic shortest path problems. In *The 27th Conference on Uncertainty in Artificial Intelligence*, Barcelona, Spain.
- Hansen, E. A. and Zilberstein, S. (2001). Lao: A heuristic search algorithm that finds solutions with loops. *Artificial Intelligence*, 129(12):35 – 62.

- Hu, H. and Liu, H. X. (2013). Arterial offset optimization using archived high-resolution traffic signal data. *Transportation Research Part C: Emerging Technologies*, 37(0):131 – 144.
- Kallenberg, L. (1983). *Linear programming and finite Markovian control problems*. Mathematisch Centrum.
- Kemeny, J. G. and Snell, J. L. (1976). *Finite Markov Chains*.
- Kirby, R. and Potts, R. (1969). The minimum route problem for networks with turn penalties and prohibitions. *Transportation Research*, 3(4):397–408.
- Kushner, H. (1971). *Introduction to stochastic control*. Holt, Rinehart and Winston.
- Liu, H., Wu, X., Ma, W., and Hu, H. (2009). Real-time queue length estimation for congested signalized intersections. *Transportation Research Part C: Emerging Technologies*, 17(4):412–427.
- Liu, H. X. and Ma, W. (2009). A virtual vehicle probe model for time-dependent travel time estimation on signalized arterials. *Transportation Research Part C: Emerging Technologies*, 17(1):11–26.
- Manne, A. S. (1960). Linear programming and sequential decisions. *Management Science*, 6(3):259–267.
- Megiddo, N. (1987). On the complexity of linear programming. In *Advances in economic theory*, pages 225–268.
- Nie, Y. M. and Li, Q. (2013). An eco-routing model considering microscopic vehicle operating conditions. *Transportation Research Part B: Methodological*, 55(0):154 – 170.

- Oneyama, H., Oguchi, T., and Kuwahara, M. (2001). Estimation model of vehicle emission considering variation of running speed. *Journal of the Eastern Asia Society for Transportation Studies*, 4(5):105–117.
- Pelkmans, L., Debal, P., Hood, T., Hauser, G., and Delgado, m.-R. (2004). Development of a simulation tool to calculate fuel consumption and emissions of vehicles operating in dynamic conditions. Technical report, SAE Technical Paper.
- Puterman, M. (1994). *Markov decision processes: Discrete stochastic dynamic programming*. John Wiley & Sons, Inc.
- Rakha, H., Ahn, K., and Moran, K. (2012). Integration framework for modeling eco-routing strategies: Logic and preliminary results. *International Journal of Transportation Science and Technology*, 1(3):259–274.
- Rakha, H., Ahn, K., and Trani, A. (2004). Development of vt-micro model for estimating hot stabilized light duty vehicle and truck emissions. *Transportation Research Part D*, 9(1):49–74.
- Rakha, H., Van Aerde, M., Ahn, K., and Trani, A. A. (2000). Requirements for evaluating traffic signal control impacts on energy and emissions based on instantaneous speed and acceleration measurements. *Transportation Research Record: Journal of the Transportation Research Board*, 1738(1):56–67.
- Tzeng, G.-H. and Chen, C.-H. (1993). Multiobjective decision making for traffic assignment. *Engineering Management, IEEE Transactions on*, 40(2):180–187.
- Whittle, P. (1983). *Optimization over time: dynamic programming and stochastic control*. Wiley series in probability and mathematical statistics. Wiley.

- Yang, B. and Miller-Hooks, E. (2004). Adaptive routing considering delays due to signal operations. *Transportation Research Part B: Methodological*, 38(5):385–413.
- Ziliaskopoulos, A. and Mahmassani, H. (1996). A note on least time path computation considering delays and prohibitions for intersection movements. *Transportation Research Part B: Methodological*, 30(5):359–367.

Appendix A

Notations

Table A.1: Notation list

Notation	Meaning
A	action space
D	destination set
$E[\cdot]$	expectation
L	link set
M	number of iteration
N	intersection set
S	state space
T	set of arrival time
a	action function
b	cost type index
c	cost function

Continued on next page

Table A.1 – continued from previous page

Notation	Meaning
d_k	decision at stage k
g	link part index
h	vehicle operation status index
i	state index
j	state index
k	stage index
l	links
m	iteration m
$p(\cdot)$	transition probability
s	state
t_v	arrival time at intersection v
t_w	arrival time at intersection w
u	upstream intersection
$u(s)$	state value of state s
v	current intersection
w	downstream intersection
x	link length
\dot{x}	free flow speed
\ddot{x}_a	acceleration rate
\ddot{x}_d	deceleration rate

Continued on next page

Table A.1 – continued from previous page

Notation	Meaning
γ	cycle length
δ	intersection delay
δ_r	red light delay
δ_q	queuing delay
ξ	offset
π	policy